Open Source Used In Appdynamics AWS Services Log Collector 24.2.0

Cisco Systems, Inc.

www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1914994996

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this **form.**

In your requests please include the following reference number 78EE117C99-1914994996

Contents

1.1 modern-go-concurrent 0.0.0-20180306012644-bacd9c7ef1dd

1.1.1 Available under license :

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License,
each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding
those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2 json-iterator-go 1.1.12

1.2.1 Available under license :

MIT License

Copyright (c) 2016 json-iterator

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION
 WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.3 modern-go-reflect2 1.0.2

1.3.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License,
each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding
those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

1.4 go-jmespath 0.4.0

1.4.1 Available under license :

MIT License

Copyright (c) 2012-2018 Mat Ryer and Tyler Bunnell

1.5 aws-aws-sdk-go-v2 1.16.8

1.5.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
```

```go
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil,
 err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription
```

```go
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
	if err !=
	nil {
		return err
	}
	err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack);
```

```
err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack,
 options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
```

}
AWS SDK for Go

```yaml
name: License Scan

on: [pull_request]

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }} ${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
```

```
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary
```

```go
	*string

	noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
```

```
    return err
  }
  if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
  }
  if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
  }
  if err = addClientUserAgent(stack, options); err != nil {
    return err
  }
  if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
  }
  if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
  }
  if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
  }
  if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return
  err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
    return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
    return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
  }
  return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
```

```
    return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "CheckInLicense",
 }
}
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
          }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n     Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
```

settings include if\n   Organizations integration has been enabled, and
which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays
the aggregated data for Linux\n   subscriptions.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
            "AmiId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
                }
            },
            "InstanceID": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
                }
            },
            "InstanceType": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The instance type of the resource.</p>"
                }

```
        },
        "AccountID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the instance.</p>"
          }
        },
        "Region": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
          }
        },
        "UsageOperation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
          }
        },
        "ProductCode": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
          "traits": {
            "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
          }
        },
        "LastUpdatedTime": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
          }
        },
        "SubscriptionName": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
          }
```

```
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
      ],

      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager Linux Subscriptions",
          "arnNamespace": "license-manager-linux-subscriptions",
          "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
```

```
        },
        "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                "x-api-key",
                "authorization",
                "x-amz-date",

    "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
            ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
                "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
                    "documentation": "The
    AWS region used to dispatch the request.",
                    "type": "String"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
```

```
            "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

          "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {

                "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }
                  ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
```

```
                {
                    "conditions": [],
                    "type": "tree",
                    "rules": [
                        {

                  "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                        "type": "error"
                    },
                    {
                        "conditions": [],

              "endpoint": {
                            "url": {
                                "ref": "Endpoint"
                            },
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    },
    {
        "conditions": [],
        "type": "tree",
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",

"argv": [
                        {
                            "ref": "Region"
```

```json
            }
          ]
        }
      ],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "type": "tree",
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
```

```json
            }
          ]
        }
      ],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "type": "tree",
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
```

```
                          "fn": "booleanEquals",
                            "argv": [
                              true,
                              {
                                "fn": "getAttr",
                                "argv": [
                                  {
                                    "ref": "PartitionResult"
                                  },
                                  "supportsFIPS"
                                ]

        }

                              ]
                          },
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              true,
                              {
                                "fn": "getAttr",
                                "argv": [
                                  {
                                    "ref": "PartitionResult"
                                  },

                              "supportsDualStack"
                                ]
                              }
                            ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
                          {
                            "conditions": [],
                            "type": "tree",
                            "rules": [
                              {

                    "conditions": [],
                                "endpoint": {
                                  "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
```

```
                                    "type": "endpoint"
                                }
                            ]
                        }
                    ]
            },
                {
                    "conditions": [],
                    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                    "type": "error"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {

            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,

    {

                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        }
```

                                ],
                                "type": "tree",

                    "rules": [
                            {
                                "conditions": [],
                                "type": "tree",
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                                            "properties": {},
                                            "headers": {}

                                        },
                                        "type": "endpoint"
                                    }
                                ]
                            }
                        ]
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled but this partition does not support FIPS",
                        "type": "error"
                    }
                ]

            },
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {

```
"conditions": [
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
],
"type": "tree",
"rules": [
    {
        "conditions": [],
        "type": "tree",
        "rules": [
            {
                "conditions": [],

                "endpoint": {
                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ]
    }
]
},

{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
]
},
{
    "conditions": [],
```

```
                                                    "type": "tree",
                                                    "rules": [
                                                        {
                                                            "conditions": [],
                                                            "endpoint": {
                                                                "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                                                "properties": {},
                                                                "headers": {}
                                                            },
                                                            "type": "endpoint"
                                                        }
                                                    ]
                                                }
                                            ]
                                        }
                                    ]
                                },
                                {
                                    "conditions": [],
                                    "error": "Invalid Configuration: Missing Region",
                                    "type": "error"
                                }
                            ]
                        }
                    ]
                },
                "smithy.rules#endpointTests": {
                    "testCases": [
                        {
                            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
                            "expect": {
                                "endpoint": {
                                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                                }
                            },
                            "params": {
                                "Region": "us-east-1",
                                "UseFIPS": true,
                                "UseDualStack": true
                            }
                        },
                        {
                            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                            "expect": {

        "endpoint": {
                                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
```

```
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {

      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
```

```
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint":

{

            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
```

```
              "Region": "us-gov-east-1",
              "UseFIPS": true,
              "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
               "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint":
{
                 "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
               "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
             "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
```

```
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
```

```
            "expect": {
               "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {

         "Region": "us-isob-east-1",
               "UseFIPS": true,
               "UseDualStack": true
            }
         },
         {
            "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
               "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
               }
            },
            "params": {
               "Region": "us-isob-east-1",
               "UseFIPS": true,
               "UseDualStack": false
            }
         },
         {
            "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
            "expect": {
               "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
               "Region": "us-isob-east-1",
               "UseFIPS": false,
               "UseDualStack": true
            }
         },
         {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
               "endpoint": {
                  "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
               }
            },
            "params": {
               "Region": "us-isob-east-1",

      "UseFIPS": false,
               "UseDualStack": false
            }
```

          },
          {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },

```
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "Missing region",
                "expect": {
                    "error": "Invalid Configuration: Missing Region"
                }

    }
        ],
        "version": "1.0"
      }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled LinuxSubscriptionsDiscovery"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
      ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
              "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type":
"operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with
commercial Linux\n      subscriptions.</p>",
          "smithy.api#http": {

      "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Instances"
          }
```

```
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by
 the name of\n        <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n        <p>The valid names for this filter
are:</p>\n      <ul>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n
</li>\n          <li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>AccountID</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n          <li>\n
   <p>\n              <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
   <code>InstanceType</code>\n            </p>\n          </li>\n        </ul>\n        <p>The valid Operators for this
filter are:</p>\n        <ul>\n          <li>\n            <p>\n              <code>contains</code>\n            </p>\n
</li>\n          <li>\n            <p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
    <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 16384
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
      }
    },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n     organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
```

```
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
          }

      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n
<li>\n          <p>\n             <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n
<p>The valid Operators for this filter are:</p>\n

      <ul>\n        <li>\n          <p>\n             <code>contains</code>\n          </p>\n        </li>\n
<li>\n          <p>\n             <code>equals</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
    <code>Notequal</code>\n          </p>\n        </li>\n      </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>",
            "smithy.api#length": {

              "min": 1,
              "max": 16384
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
      }
    },
```

```json
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},

"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
```

```
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [

{
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits":
{
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"

            }
        },
        "Type": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n     Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n     Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
            }
        },
```

```
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },

        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
              "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
              },
              "smithy.api#idempotent": {}
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
  {
              "type": "structure",
              "members": {
                "LinuxSubscriptionsDiscovery": {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                  "traits": {
                    "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                    "smithy.api#required": {}
                  }
                },
                "LinuxSubscriptionsDiscoverySettings": {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                  "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",

      "smithy.api#required": {}
                }
              },
              "AllowUpdate": {
                "target": "smithy.api#Boolean",
                "traits": {
                  "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                }
              }
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
            "type": "structure",
            "members": {
              "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },

      "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
          "traits": {
            "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
          }
        },
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n     subscriptions.</p>"
          }
        }
      }
    }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
      }
    }
```

```go
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //   - Beneficiary
 //   - ProductSKU
 //   - Fingerprint
 //
   - Status
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
```

```go
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if
    err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
```

```
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
```

```go
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
    ServiceID,
  OperationName: "ListLicenses",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
```

//
//
This member is required.
LicenseType types.LicenseType

// The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
 }
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
 }
if err := addProtocolFinalizerMiddlewares(stack,
options, "DisassociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
 }

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
 }
if err = addSetLoggerMiddleware(stack, options); err != nil {

```
  return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
```

```go
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
	{
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "DisassociateLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
	if params == nil {
		params = &CheckoutLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
	if err != nil {
```

```go
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type
CheckoutLicenseInput struct {

    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {
```

```go
	// Checkout type.
	CheckoutType types.CheckoutType

	// Allowed license entitlements.
	EntitlementsAllowed []types.EntitlementData

	// Date and time at which the license checkout expires.
	Expiration *string

	// Date and time at which the license checkout is issued.
	IssuedAt *string

	// Amazon Resource
	Name (ARN) of the checkout license.
	LicenseArn *string

	// License consumption token.
	LicenseConsumptionToken *string

	// Node ID.
	NodeId *string

	// Signed token.
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil
```

```go
{
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack);
err != nil {
 return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
```

```go
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
	if params == nil {
		params = &CheckoutBorrowLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
```

```go
	if err != nil {
	 return nil, err
	}

	out := result.(*CheckoutBorrowLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type CheckoutBorrowLicenseInput struct {

	// Unique, case-sensitive identifier
	that you provide to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// Digital signature method. The possible value is JSON Web Signature (JWS)
	// algorithm PS384. For more information, see RFC 7518 Digital Signature with
	// RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
	//
	// This member is required.
	DigitalSignatureMethod types.DigitalSignatureMethod

	// License entitlements. Partial checkouts are not supported.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Amazon Resource Name (ARN) of the license. The license must use the borrow
	// consumption configuration.
	//
	// This member is required.
	LicenseArn *string

	// Information about constraints.
	CheckoutMetadata []types.Metadata

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

	// Information about constraints.
```

CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return
err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
```

```
  return err
  }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
  }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
  }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
  }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
  }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
```

```go
	if err != nil {
		return nil, err
	}

	out := result.(*CreateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type CreateLicenseInput struct {

	// License beneficiary.
	//
	// This member is required.
	Beneficiary *string

	// Unique, case-sensitive identifier that you provide
	to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// Configuration for consumption of the license. Choose a provisional
	// configuration for workloads running with continuous connectivity. Choose a
	// borrow configuration for workloads with offline usage.
	//
	// This member is required.
	ConsumptionConfiguration *types.ConsumptionConfiguration

	// License entitlements.
	//
	// This member is required.
	Entitlements []types.Entitlement

	// Home Region for the license.
	//
	// This member is required.
	HomeRegion *string

	// License issuer.
	//
	// This member is required.
	Issuer *types.Issuer

	// License name.
	//
	// This member is required.
```

```go
	LicenseName *string

	// Product name.
	//
	// This member is required.
	ProductName *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// Date and time range during which the license is valid, in ISO8601-UTC format.
	//
	// This member is required.
	Validity
	*types.DatetimeRange

	// Information about the license.
	LicenseMetadata []types.Metadata

	noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
```

```
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add
protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
```

```
    return err
  }
  if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
   return err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
   return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
   return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
   return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
   return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
   return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
   return err
  }
  return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
  return &awsmiddleware.RegisterServiceMetadata{
   Region:        region,
   ServiceID:     ServiceID,
   OperationName:
  "CreateLicense",
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```go
// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
	if params == nil {
		params = &GetLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*GetLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type GetLicenseInput struct {

	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// License version.
	Version
	*string

	noSmithyDocumentSerde
}

type GetLicenseOutput struct {

	// License details.
	License *types.License

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
```

```
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return
err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpGetLicenseValidationMiddleware(stack);
    err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
    err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:        region,
        ServiceID:     ServiceID,
        OperationName: "GetLicense",
    }
}
```

distribution.
 * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```go
type ListReceivedLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//
	//   - ProductSKU
	//   - Status
	//   - Fingerprint
	//   - IssuerName
	//   - Beneficiary
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

	// Received license details.
	Licenses []types.GrantedLicense

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
```

```
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack);
err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
```

```
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "ListReceivedLicenses",
 }
}
```

<div align="center">

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

</div>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
 incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
                "smithy.api#error": "client"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
            "type": "structure",
            "members": {
                "DirectoryId": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation":
"<p>Details about an Active Directory identity provider.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
```

```
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related
costs will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
      monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/AssociateUser"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
```

```
        "traits": {
          "smithy.api#documentation": "<p>The identity provider of the user.</p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the user.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n     resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/DeregisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }

    },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            }
          }
```

```
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },

    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
          "type": "structure",
          "members": {
            "Attribute": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
              }
            },
            "Operation": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
              }
            },
            "Value": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used
to return more specific results from a describe\n     operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n     attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n    settings such as the subnets to provision VPC endpoints.</p>",
            "smithy.api#required":
{}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
```

```
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status
 of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
```

```
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                }
            }

        }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits":
{
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
           "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        },
        "AssociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
          }
        },
        "DisassociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
          }
        }

  },
      "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",

      "version": "2018-05-10",
        "operations": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
          },

  },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
```

```
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"

  ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
```

```
                    "documentation": "When true, use the dual-stack
endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                    "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }

            },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },

                                        true
                                    ]
                                }
                            ],
                            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
```

```
                    "type": "error"
                },
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],

"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],
                    "endpoint": {
                        "url": {
                            "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [

{
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
```

```
                            "fn": "aws.partition",
                            "argv": [
                               {
                                  "ref": "Region"
                               }
                            ],
                            "assign": "PartitionResult"
                         }
                      ],
                      "type": "tree",

         "rules": [
                      {
                         "conditions": [
                            {
                               "fn": "booleanEquals",
                               "argv": [
                                  {
                                     "ref": "UseFIPS"
                                  },
                                  true
                               ]
                            },
                            {
                               "fn": "booleanEquals",
                               "argv": [
                                  {
                                     "ref": "UseDualStack"

                                  },
                                  true
                               ]
                            }
                         ],
                         "type": "tree",
                         "rules": [
                            {
                               "conditions": [
                                  {
                                     "fn": "booleanEquals",
                                     "argv": [
                                        true,
                                        {
                                           "fn": "getAttr",

"argv": [

                                           {
                                              "ref": "PartitionResult"
```

                                                          },
                                                          "supportsFIPS"
                                                      ]
                                                  }
                                              ]
                                          },
                                          {
                                              "fn": "booleanEquals",
                                              "argv": [
                                                  true,
                                                  {
                        "fn": "getAttr",
                                                      "argv": [
                                                          {
                                                              "ref": "PartitionResult"
                                                          },
                                                          "supportsDualStack"
                                                      ]
                                                  }
                                              ]
                                          }
                                      ],
                                      "type": "tree",
                                      "rules": [
                                          {
                        "conditions": [],
                                              "endpoint": {
                                                  "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                  "properties": {},
                                                  "headers": {}
                                              },
                                              "type": "endpoint"
                                          }
                                      ]
                                  },
                                  {
                                      "conditions": [],
                                      "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                                       "type": "error"
                                  }
                              ]
                          },
                          {
                              "conditions": [

```
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseFIPS"
                            },
                            true
                          ]
                        }
                      ],
                      "type": "tree",

          "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
                                      "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"

                                  ]
                                }
                              ]
                            }
                          ],
                          "type": "tree",
                          "rules": [
                            {
                              "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}

          },
                              "type": "endpoint"
                            }
                          ]
                        },
                        {
```

```
                        "conditions": [],
                        "error": "FIPS is enabled but this partition does not support FIPS",
                        "type": "error"
                    }
                ]
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {

                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {

                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
```

```
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ]
                },
                {
                    "conditions": [],
                    "error": "DualStack is enabled but this partition does
not support DualStack",
                    "type": "error"
                }
            ]
        },
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ]
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
```

```
                    "UseDualStack": true
                }
            },
            {
                "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                    }
                },

        "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
                "expect": {
```

```
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },

      "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
```

```
              "UseFIPS": false,
              "UseDualStack": false
          }
      },
      {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
              "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
              }
          },

  "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": true,
              "UseDualStack": true
          }
      },
      {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
              "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
              }
          },
          "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": true,
              "UseDualStack": false
          }
      },
      {
          "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
          "expect": {
              "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
              }
          },
          "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": true
          }
      },
      {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
```

```
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
              }
            },
      "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
            "expect":
{
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack":
true
            }
          },
```

```
            {
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                    }
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }

            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled but this partition does not support DualStack"
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
```

```
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
```

```
                    "params": {
                        "Region": "us-east-1",
                         "UseFIPS": true,
                        "UseDualStack": false,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
      },
      "smithy.api#paginated":
{
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
```

```
{
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
```

```
            ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type":
"structure",
        "members": {
            "InstanceSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
```

```
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
```

```
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {

  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n    more sets of key-value pairs that you specify.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    }
```

```
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
              }
            },

      "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
```

```
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceUserSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
```

```
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {

            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }

        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for a user.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
          }
        },
        "SubscriptionEndDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/RegisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n      subnets to provision VPC endpoints.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
        "type": "string",

  "traits": {
        "smithy.api#length": {
          "min": 5,
          "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
      }
    },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n     your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
```

```
          "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n     subnets to provision VPC endpoints, and the security group ID that
 is associated with the VPC\n     endpoints. The security group should permit inbound TCP port 1688
communication from resources\n     in the VPC.</p>"
          }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
             },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n     <note>\n          <p>Your estimated bill for charges on the number of users and related costs
will take 48\n       hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
 For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n        monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
            "smithy.api#http": {
                "code": 200,
```

```
              "method": "POST",
              "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
```

```
                        "target": "smithy.api#String",
                        "traits": {
                          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                          "smithy.api#required": {}
                        }
                      },
                      "IdentityProvider": {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                        "traits": {
                          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                          "smithy.api#required":
{}
                        }
                      },
                      "Product": {
                        "target": "smithy.api#String",
                        "traits": {
                          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                          "smithy.api#required": {}
                        }
                      },
                      "Domain": {
                        "target": "smithy.api#String",
                        "traits": {
                          "smithy.api#documentation": "<p>The domain name of the user.</p>"
                        }
                      }
                    }
                  }
                },
                "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
                  "type": "structure",
                  "members": {
                    "ProductUserSummary": {
                      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                      "traits": {
                        "smithy.api#documentation":
"<p>Metadata that describes the start product subscription operation.</p>",
                        "smithy.api#required": {}
                      }
                    }
                  }
                },
                "com.amazonaws.licensemanagerusersubscriptions#StringList": {
                  "type": "list",
                  "member": {
                    "target": "smithy.api#String"
                  }
                },
```

```json
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
    "type": "structure",
    "members": {
        "message": {

            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
```

```
      "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity\n      provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
           "traits": {
              "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
          }
        },
        "UpdateSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
          "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n      update any combination of settings in a single operation such as the:</p>\n
<ul>\n      <li>\n            <p>Subnets which you want to add to provision VPC endpoints.</p>\n          </li>\n
     <li>\n            <p>Subnets which you want to remove the VPC endpoints from.</p>\n          </li>\n
<li>\n            <p>Security group ID which permits traffic to the VPC endpoints.</p>\n          </li>\n      </ul>",
              "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
              "smithy.api#required": {}
          }
```

```
              }
            }
          },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
          "type": "structure",
          "members": {
            "AddSubnets": {

              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                  "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n     require connectivity to activation servers.</p>",
                  "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
              }
            },
            "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                "smithy.api#documentation": "<p>A security group ID
that allows inbound TCP port 1688 communication between resources in\n     your VPC and the VPC endpoints for
activation servers.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as\n     the subnets to provision VPC endpoints.</p>"
          }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
          }
      }
```

```
          }
       }
       {
          "smithy": "2.0",
          "metadata": {
             "suppressions": [
                {
                   "id": "HttpMethodSemantics",
                   "namespace": "*"
                },
                {
                   "id": "HttpResponseCodeSemantics",
                   "namespace": "*"
                },
                {
                   "id": "PaginatedTrait",
                   "namespace": "*"
                },
                {
                   "id": "HttpHeaderTrait",
                   "namespace": "*"
                },
                {
                   "id": "HttpUriConflict",
                   "namespace": "*"
                },
                {
                   "id": "Service",
                   "namespace": "*"
                }
             ]
          },
          "shapes": {
             "com.amazonaws.licensemanager#AWSLicenseManager": {
                "type": "service",
                "version": "2018-08-01",
                "operations": [
                   {
                      "target": "com.amazonaws.licensemanager#AcceptGrant"
                   },

       {
                      "target": "com.amazonaws.licensemanager#CheckInLicense"
                   },
                   {
                      "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
                   },
                   {
```

      "target": "com.amazonaws.licensemanager#CheckoutLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateGrantVersion"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
     },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateToken"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
      "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    },
},
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },

```
{
    "target": "com.amazonaws.licensemanager#GetGrant"
},
{
    "target": "com.amazonaws.licensemanager#GetLicense"
},
{
    "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
},
{
    "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
},
{
    "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
},
{
    "target": "com.amazonaws.licensemanager#GetLicenseUsage"
},
{
    "target": "com.amazonaws.licensemanager#GetServiceSettings"
},
{
    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
},
{
    "target": "com.amazonaws.licensemanager#ListDistributedGrants"
},
{
    "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenses"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
},
{
```

      "target": "com.amazonaws.licensemanager#ListLicenseVersions"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListResourceInventory"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTokens"
    },
    {
      "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
     },
    {
      "target": "com.amazonaws.licensemanager#RejectGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
    }
  ],

```
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager",
        "arnNamespace": "license-manager",
        "cloudFormationName": "LicenseManager",
        "cloudTrailEventSource": "licensemanager.amazonaws.com",
        "endpointPrefix": "license-manager"
    },
    "aws.auth#sigv4": {
        "name": "license-manager"
    },
    "aws.protocols#awsJson1_1": {},
    "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n        Amazon Web Services accounts and on-premises servers.</p>",
    "smithy.api#title": "AWS License Manager",
    "smithy.api#xmlNamespace": {
        "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
    },
    "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
            "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "Boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
                "type": "Boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
```

```
                    "type": "String"
                }
            },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]

                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },
                                        true
                                    ]
                                }
                            ],
                            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                            "type": "error"
                        },
                        {

"conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
                                        },
                                        true
                                    ]
                                }
                            ],
                            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                            "type": "error"
```

```
      },
      {
        "conditions": [],
        "endpoint": {
          "url": {
            "ref": "Endpoint"
          },
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ]
  },
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "type": "tree",
    "rules": [
      {
        "conditions": [
          {
            "fn": "aws.partition",
            "argv": [
              {
                "ref": "Region"
              }
            ],
            "assign": "PartitionResult"
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref":
```

"UseFIPS"

```
              },
              true
            ]
          },
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  true,
                  {
                    "fn": "getAttr",
                    "argv": [
                      {
                        "ref": "PartitionResult"
                      },
                      "supportsFIPS"
                    ]
                  }
                ]
              }
            ]
          },
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
```

"supportsDualStack"

```
                ]
```

```
                            }
                          ]
                        }
                      ],
                      "type": "tree",
                      "rules": [
                        {
                          "conditions": [],
                          "endpoint": {
                            "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}

                          },
                            "type": "endpoint"
                        }
                      ]
                    },
                    {
                      "conditions": [],
                      "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                      "type": "error"
                    }
                  ]
                },
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",

                  "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
```

```
                    {
                          "fn": "getAttr",
                          "argv": [
                            {
                               "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                       }
                    ]
                 }
              ],
              "type": "tree",
              "rules": [

                 {
                       "conditions": [],
                       "endpoint": {
                          "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                          "properties": {},
                          "headers": {}
                       },
                       "type": "endpoint"
                    }
                 ]
              },
              {
                 "conditions": [],
                 "error": "FIPS is enabled but this
partition does not support FIPS",
                 "type": "error"
              }
           ]
        },
        {
           "conditions": [
              {
                 "fn": "booleanEquals",
                 "argv": [
                    {
                       "ref": "UseDualStack"
                    },
                    true
                 ]
              }
           ],
           "type": "tree",
```

```
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ],
    "type": "tree",
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ]
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
]
},
{
  "conditions": [],
  "endpoint": {
    "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
```

```
                  },
                  "type": "endpoint"
              }
            ]
          }
        ]
      },
      {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
      }
    ]
  },
  "smithy.rules#endpointTests": {
    "testCases": [
      {
        "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint":
{
            "url": "https://license-manager.af-south-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "af-south-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-1.amazonaws.com"
```

```
              }
            },
            "params": {
              "Region": "ap-northeast-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.ap-northeast-2.amazonaws.com"
              }
            },
            "params": {
              "Region": "ap-northeast-2",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.ap-northeast-3.amazonaws.com"
              }
            },
            "params": {
              "Region": "ap-northeast-3",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.ap-south-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "ap-south-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
```

```
{
  "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.ap-southeast-1.amazonaws.com"

    }
  },
  "params": {
    "Region": "ap-southeast-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.ap-southeast-2.amazonaws.com"
    }
  },
  "params": {
    "Region": "ap-southeast-2",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{

  "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.ap-southeast-3.amazonaws.com"
    }
  },
  "params": {
    "Region": "ap-southeast-3",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.ca-central-1.amazonaws.com"
    }
  },
```

```
          "params": {
              "Region": "ca-central-1",
              "UseFIPS": false,
              "UseDualStack": false
          }
      },
      {
          "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager.eu-central-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-central-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
      },
      {
          "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager.eu-north-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
      },
      {
          "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager.eu-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region":
"eu-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
      },
```

```
{
    "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.eu-west-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "eu-west-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
    "expect": {

        "endpoint": {
            "url": "https://license-manager.eu-west-2.amazonaws.com"
        }
    },
    "params": {
        "Region": "eu-west-2",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.eu-west-3.amazonaws.com"
        }
    },
    "params": {
        "Region": "eu-west-3",
        "UseFIPS": false,

"UseDualStack": false
    }
},
{
    "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.me-south-1.amazonaws.com"
        }
    },
```

```
        "params": {
          "Region": "me-south-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.sa-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "sa-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }

      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
```

```
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-east-2.amazonaws.com"
            }
          }
        },
        "params": {
          "Region": "us-east-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-east-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-2",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region
us-west-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-west-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-west-1.amazonaws.com"
          }
        },
        "params": {
```

```
        "Region": "us-west-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-west-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-west-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

"expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-west-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-west-2",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,

            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-northwest-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    }
  }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
```

```
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },

        {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-gov-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
```

```
            "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-west-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-west-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-gov-east-1.api.aws"
          }
        },

    "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": true
```

```
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
```

```
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": false
          }
      },
      {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
              "error": "DualStack is enabled
but this partition does not support DualStack"
          },
          "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": true
          }
      },
      {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
              "endpoint": {
                  "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
              }
          },
          "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": false,
              "UseDualStack": false
          }
      },

      {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": true
          }
      },
      {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
              "endpoint": {
                  "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
              }
```

```
                    },

        "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

    "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,

    "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
```

```
              },
              {
                 "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                 "expect": {
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                 },
                 "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                 }
              },
              {
                 "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                 "expect": {
                    "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
                 },
                 "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                 }
              },
              {
                 "documentation": "Missing region",
                 "expect": {
                    "error": "Invalid Configuration: Missing Region"
                 }
              }
           ],
           "version": "1.0"
        }
     }
  },
  "com.amazonaws.licensemanager#AcceptGrant": {
     "type": "operation",
     "input": {
        "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

},
     "output": {
        "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
     },
     "errors": [
        {
```

```
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
        }
    },
    "com.amazonaws.licensemanager#AcceptGrantRequest": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#AcceptGrantResponse": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Grant
```

```
ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#AccessDeniedException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {

  "code": "ServiceAccessDenied",
          "httpResponseCode": 401
        },
        "smithy.api#documentation": "<p>Access to resource denied.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 401
      }
    },
    "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
      "type": "enum",
      "members": {
        "DISTRIBUTED_GRANTS_ONLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
          }
        },
        "ALL_GRANTS_PERMITTED_BY_ISSUER": {
```

```
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
              }
            }
          }
        }
      },
       "com.amazonaws.licensemanager#AllowedOperation": {

    "type": "enum",
        "members": {
          "CREATE_GRANT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CreateGrant"
            }
          },
          "CHECKOUT_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CheckoutLicense"
            }
          },
          "CHECKOUT_BORROW_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CheckoutBorrowLicense"
            }
          },
          "CHECK_IN_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CheckInLicense"
            }
          },
          "EXTEND_CONSUMPTION_LICENSE":
{
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "ExtendConsumptionLicense"
              }
            },
          "LIST_PURCHASED_LICENSES": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ListPurchasedLicenses"
            }
          },
```

```
          "CREATE_TOKEN": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CreateToken"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#AllowedOperationList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#AllowedOperation"
        },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max":
7
        }
      }
      },
      "com.amazonaws.licensemanager#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 2048
          },
          "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
        }
      },
      "com.amazonaws.licensemanager#ArnList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Arn"
        }
      },
      "com.amazonaws.licensemanager#AuthorizationException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError":
```

```
  {
            "code": "AuthorizationFailure",
            "httpResponseCode": 403
          },
          "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n        policy associated with this account.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 403
        }
      },
      "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
        "type": "structure",
        "members": {
          "LastRunTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes automated discovery.</p>"
        }
      },

  "com.amazonaws.licensemanager#Boolean": {
        "type": "boolean",
        "traits": {
          "smithy.api#default": false
        }
      },
      "com.amazonaws.licensemanager#BorrowConfiguration": {
        "type": "structure",
        "members": {
          "AllowEarlyCheckIn": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxTimeToLiveInMinutes": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
              "smithy.api#required": {}
            }
          }
```

```
        },
            "traits": {
                "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#BoxBoolean": {
            "type": "boolean"
        },
        "com.amazonaws.licensemanager#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanager#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanager#CheckInLicense": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
          }
        },
        "com.amazonaws.licensemanager#CheckInLicenseRequest":
 {
            "type": "structure",
            "members": {
              "LicenseConsumptionToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>License consumption token.</p>",
                  "smithy.api#required": {}
                }
              },
              "Beneficiary": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>License beneficiary.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#CheckInLicenseResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#CheckoutBorrowLicense":
 {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
          },
          {
            "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
        "type": "structure",
        "members":
{
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
              "smithy.api#required": {}
            }
```

```
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
            "smithy.api#required": {}
          }
        },
        "DigitalSignatureMethod": {
          "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
          "traits": {
            "smithy.api#documentation": "<p>Digital signature method. The possible
 value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
            "smithy.api#required": {}
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
                }
            },
            "LicenseConsumptionToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License consumption token.</p>"
                }
            },
            "EntitlementsAllowed":
{
                "target": "com.amazonaws.licensemanager#EntitlementDataList",
                "traits": {
                    "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
                }
            },
            "NodeId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Node ID.</p>"
                }
            },
            "SignedToken": {
                "target": "com.amazonaws.licensemanager#SignedToken",
                "traits": {
                    "smithy.api#documentation": "<p>Signed token.</p>"
                }
            },
            "IssuedAt": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
                }
            },
            "Expiration": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
                }
            },
            "CheckoutMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Information about constraints.</p>"
                }
            }
        },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
       },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n          <p>If the
account that created the license is the same that is performing the check out, you must\n          specify the account
```

as the beneficiary.</p>\n   </note>"
        }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
        "type": "structure",
        "members": {
            "ProductSKU": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Product
SKU.</p>",
                    "smithy.api#required": {}
                }
            },
            "CheckoutType": {
                "target": "com.amazonaws.licensemanager#CheckoutType",
                "traits": {
                    "smithy.api#documentation": "<p>Checkout type.</p>",
                    "smithy.api#required": {}
                }
            },
            "KeyFingerprint": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
                    "smithy.api#required": {}
                }
            },
            "Entitlements": {
                "target": "com.amazonaws.licensemanager#EntitlementDataList",
                "traits": {
                    "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
                }
            },
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            },
            "Beneficiary": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License beneficiary.</p>"

```
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          }
        }
      },
      "traits": {

    "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
      "type": "structure",
      "members": {
        "CheckoutType": {
          "target": "com.amazonaws.licensemanager#CheckoutType",
          "traits": {
            "smithy.api#documentation": "<p>Checkout type.</p>"
          }
        },
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },
        "EntitlementsAllowed": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
          }
        },
         "SignedToken": {
          "target": "com.amazonaws.licensemanager#SignedToken",
          "traits": {
            "smithy.api#documentation": "<p>Signed token.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "IssuedAt": {
```

```
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
            }
          },
          "Expiration": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutType": {
        "type": "enum",
        "members": {
          "PROVISIONAL": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PROVISIONAL"
            }
          },
          "PERPETUAL": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PERPETUAL"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#ClientRequestToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 36
          }
        }
```

```
        },
        "com.amazonaws.licensemanager#ClientToken": {
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 2048
            },
            "smithy.api#pattern": "^\\S+$"
          }
        },
        "com.amazonaws.licensemanager#ConflictException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "ConflictException",
              "httpResponseCode":
409
            },
            "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 409
          }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
          "type": "structure",
          "members": {
            "ResourceType": {
              "target": "com.amazonaws.licensemanager#ResourceType",
              "traits": {
                "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
              }
            },
            "ConsumedLicenses": {
              "target": "com.amazonaws.licensemanager#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
              }
            }
          },
```

```
"traits": {
        "smithy.api#documentation": "<p>Details about license consumption.</p>"
    }
},
"com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
    }
},
"com.amazonaws.licensemanager#ConsumptionConfiguration": {
    "type": "structure",
    "members": {
        "RenewType": {
            "target": "com.amazonaws.licensemanager#RenewType",
            "traits": {
                "smithy.api#documentation": "<p>Renewal frequency.</p>"
            }
        },
        "ProvisionalConfiguration": {
            "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
            }
        },
        "BorrowConfiguration": {
            "target": "com.amazonaws.licensemanager#BorrowConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
    }
},
"com.amazonaws.licensemanager#CreateGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
```

```
                {
                    "target":
"com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "com.amazonaws.licensemanager#CreateGrantRequest": {
            "type": "structure",
            "members": {
                "ClientToken": {
                    "target": "com.amazonaws.licensemanager#ClientToken",
                    "traits": {
                        "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                        "smithy.api#required": {}
                    }
                },
                "GrantName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant name.</p>",
                        "smithy.api#required": {}

    }
                },
                "LicenseArn": {
```

```
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                "smithy.api#required": {}
              }
            },
            "Principals": {
              "target": "com.amazonaws.licensemanager#PrincipalArnList",
              "traits": {
                "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n        (ARN):</p>\n        <ul>\n          <li>\n              <p>An Amazon Web Services
account, which includes only the account specified.</p>\n          </li>\n        </ul>\n        <ul>\n          <li>\n
    <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n          </li>\n        </ul>\n
<ul>\n
        <li>\n              <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n        </ul>",
                "smithy.api#required": {}
              }
            },
            "HomeRegion": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowedOperations": {
              "target": "com.amazonaws.licensemanager#AllowedOperationList",
              "traits": {
                "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },

        "com.amazonaws.licensemanager#CreateGrantResponse": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Grant ARN.</p>"
              }
            },
```

```
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateGrantVersion":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
```

```
        ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
      "type": "structure",
      "members": {
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
        }
      },
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant name.</p>"
        }
      },
      "AllowedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
        "traits": {
          "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#GrantStatus",
         "traits": {
          "smithy.api#documentation": "<p>Grant status.</p>"
        }
      },
      "StatusReason": {
        "target": "com.amazonaws.licensemanager#StatusReasonMessage",
        "traits": {
            "smithy.api#documentation": "<p>Grant status reason.</p>"
```

```
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the grant.</p>"
            }
          },
          "Options": {
            "target": "com.amazonaws.licensemanager#Options",
            "traits": {
              "smithy.api#documentation": "<p>The options specified for the grant.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    }
  },
  "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Grant ARN.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#GrantStatus",
        "traits": {
          "smithy.api#documentation": "<p>Grant status.</p>"
        }
      },
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>New version of the grant.</p>"
        }
      }
    },
    "traits":
{
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#CreateLicense": {
    "type": "operation",
```

    "input": {
      "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a license.</p>"
    }
  },
  "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"

```
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
 all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of
licenses purchased and used.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Name of the license configuration.</p>",
                    "smithy.api#required": {}
                }
            },
            "Description": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Description of the license configuration.</p>"
                }
            },
            "LicenseCountingType": {
                "target": "com.amazonaws.licensemanager#LicenseCountingType",
                "traits": {
                    "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseCount": {
                "target": "com.amazonaws.licensemanager#BoxLong",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
            }
          },
          "LicenseCountHardLimit": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n        blocks the launch of new instances.</p>"
            }
          },
          "LicenseRules": {
            "target":
 "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n        vary by dimension, as follows.</p>\n
<ul>\n         <li>\n            <p>\n               <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
      <code>licenseAffinityToHost</code> |\n               <code>maximumCores</code> |
<code>minimumCores</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n            <code>maximumCores</code> |
<code>minimumCores</code> |\n            <code>maximumSockets</code> | <code>minimumSockets</code> |\n
      <code>maximumVcpus</code> | <code>minimumVcpus</code>\n            </p>\n         </li>\n
<li>\n
  <p>\n               <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n               <code>maximumSockets</code> |
<code>minimumSockets</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n            <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n            </p>\n         </li>\n      </ul>\n         <p>The unit for
<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n        values for
<code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n
<code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n
<code>True</code> and <code>False</code>.</p>"
            }
          },

    "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
            }
          },
          "DisassociateWhenNotFound": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
```

```
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>Product information.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
         },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {

          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
            "smithy.api#required": {}
          }
        },
        "SourceLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n       For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
            "smithy.api#required": {}
          }
        },
        "DestinationLicenseContext":
 {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

            "traits": {
                "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
```

```
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "ReportGeneratorName":
{
          "target": "com.amazonaws.licensemanager#ReportGeneratorName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>",
            "smithy.api#required": {}
          }
        },
        "Type": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n    <ul>\n       <li>\n          <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n       </li>\n       <li>\n          <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n       </li>\n
</ul>",
            "smithy.api#required": {}


        }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
            "smithy.api#required": {}
          }
        },
        "ReportFrequency": {
          "target": "com.amazonaws.licensemanager#ReportFrequency",
          "traits": {
            "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
            "smithy.api#required": {}
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientRequestToken",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Unique,
case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
```

                },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product
name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product SKU.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits": {
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region for the license.</p>",
              "smithy.api#required": {}

    }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
              "smithy.api#required": {}
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",

```
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>",
              "smithy.api#required": {}
            }
          },
          "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n       running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n       usage.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license.</p>"
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
```

```
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License version.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#CreateLicenseVersion": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#RedirectException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
  }
```

```
        },
      "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target":
"com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product name.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits":
{
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
              "smithy.api#required": {}
            }
```

```
        },
        "LicenseMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license.</p>"
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",
            "smithy.api#required": {}
          }
        },
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
            "smithy.api#required": {}
          }
        },

    "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>",
            "smithy.api#required": {}
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        },
        "SourceVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the license.</p>"
          }
        }
      },
      "traits": {
```

```
"smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>License ARN.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>New version of the license.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#LicenseStatus",
                "traits": {
                    "smithy.api#documentation": "<p>License status.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output":
{}
        }
    },
    "com.amazonaws.licensemanager#CreateToken": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateTokenRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateTokenResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },


    {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateTokenRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits":
 {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
            "smithy.api#required": {}
          }
        },
        "RoleArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n        License Manager does not check whether the roles are in use.</p>"
          }
        },
        "ExpirationInDays": {
          "target": "com.amazonaws.licensemanager#Integer",
          "traits": {
            "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
          }
```

```
        },
        "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
            "traits": {
              "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n      to the amr claim of the JWT token.</p>"
            }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#CreateTokenResponse": {
    "type": "structure",
    "members": {
      "TokenId": {
        "target": "com.amazonaws.licensemanager#String",

        "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
      },
      "TokenType": {
        "target": "com.amazonaws.licensemanager#TokenType",
        "traits": {
          "smithy.api#documentation": "<p>Token type.</p>"
        }
      },
      "Token": {
        "target": "com.amazonaws.licensemanager#TokenString",
        "traits": {
          "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#DateTime": {
```

```
      "type": "timestamp"
    },
    "com.amazonaws.licensemanager#DatetimeRange": {
      "type": "structure",
      "members": {
        "Begin": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Start of the time range.</p>",
            "smithy.api#required": {}
          }
        },
        "End": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>End of the time range.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
        }
    },
    "com.amazonaws.licensemanager#DeleteGrantRequest": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "StatusReason": {
                "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                "traits": {
                    "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Current version of the
grant.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteGrantResponse": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
```

```
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation":
"<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
        },

      "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
        "type": "structure",
        "members":
{
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>ID of the license configuration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
```

```
                }
              ],
              "traits": {
                "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
              }
            },
            "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
              "type": "structure",
              "members": {
                "LicenseManagerReportGeneratorArn": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
                    "smithy.api#required": {}
                  }
                }
              },
              "traits": {

  "smithy.api#input": {}
              }
            },
            "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
              "type": "structure",
              "members": {},
              "traits": {
                "smithy.api#output": {}
              }
            },
            "com.amazonaws.licensemanager#DeleteLicenseRequest": {
              "type": "structure",
              "members": {
                "LicenseArn": {
                  "target": "com.amazonaws.licensemanager#Arn",
                  "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                    "smithy.api#required": {}
                  }
                },
                "SourceVersion": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Current version of the license.</p>",
                    "smithy.api#required":
```

```
{}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseResponse": {
      "type": "structure",
      "members": {
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "DeletionDate": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteToken": {
      "type": "operation",
       "input": {
        "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
```

```
          },
          {
              "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
          },

    {
              "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteTokenRequest": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteTokenResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}

     }
      },
      "com.amazonaws.licensemanager#DigitalSignatureMethod": {
        "type": "enum",
        "members": {
          "JWT_PS384": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "JWT_PS384"
            }
```

```
          }
        }
      },
      "com.amazonaws.licensemanager#Entitlement": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement name.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
            }
          },
          "MaxCount": {
            "target": "com.amazonaws.licensemanager#Long",
            "traits": {
              "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
            }
          },
          "Overage": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement unit.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowCheckIn": {
            "target":
"com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
            }
          }
        },
        "traits": {
```

```
                    "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
                }
            },
            "com.amazonaws.licensemanager#EntitlementData": {
                "type": "structure",
                "members": {
                    "Name": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Entitlement data name.</p>",
                            "smithy.api#required": {}
                        }
                    },
                    "Value": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Entitlement
data value.</p>"
                        }
                    },
                    "Unit": {
                        "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
                        "traits": {
                            "smithy.api#documentation": "<p>Entitlement data unit.</p>",
                            "smithy.api#required": {}
                        }
                    }
                },
                "traits": {
                    "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
                }
            },
            "com.amazonaws.licensemanager#EntitlementDataList": {
                "type": "list",
                "member": {
                    "target": "com.amazonaws.licensemanager#EntitlementData"
                }
            },
            "com.amazonaws.licensemanager#EntitlementDataUnit": {
                "type": "enum",
                "members": {
                    "COUNT": {
                        "target": "smithy.api#Unit",
                        "traits": {

"smithy.api#enumValue": "Count"
                        }
                    },
```

```
"NONE": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "None"
  }
},
"SECONDS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Seconds"
  }
},
"MICROSECONDS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Microseconds"
  }
},
"MILLISECONDS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Milliseconds"
  }
},
"BYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Bytes"
  }
},
"KILOBYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Kilobytes"
  }
},
"MEGABYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Megabytes"
  }
},
"GIGABYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Gigabytes"
  }
},
```

```
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS":
{

      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits"
      }
    },
    "KILOBITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobits"
      }
    },
    "MEGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits"
      }
    },
    "GIGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabits"
      }
    },
    "TERABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabits"

}
    },
    "PERCENT": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Percent"
      }
    },
    "BYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes/Second"
```

```
        }
      },
      "KILOBYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes/Second"
        }
      },
      "MEGABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes/Second"
        }
      },
      "GIGABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes/Second"
        }
      },
      "TERABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabytes/Second"
        }
      },
      "BITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits/Second"
        }
      },
      "KILOBITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits/Second"
        }
      },
      "MEGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue":
"Megabits/Second"
        }
      },
      "GIGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
```

```
              "smithy.api#enumValue": "Gigabits/Second"
            }
          },
          "TERABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Terabits/Second"
            }
          },
          "COUNT_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count/Second"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#EntitlementList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Entitlement"
        }
      },
      "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#EntitlementUnit": {
        "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
            "target": "smithy.api#Unit",
```

```
        "traits": {
          "smithy.api#enumValue": "None"
        }
      },
      "SECONDS":
{
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Seconds"
        }
      },
      "MICROSECONDS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Microseconds"
        }
      },
      "MILLISECONDS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Milliseconds"
        }
      },
      "BYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes"
        }
      },
      "KILOBYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes"
         }
      },
      "MEGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes"
        }
      },
      "GIGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes"
        }
      },
      "TERABYTES": {
```

```json
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabytes"
          }
        },
        "BITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits"
          }
        },
        "KILOBITS": {
          "target": "smithy.api#Unit",
          "traits": {

            "smithy.api#enumValue": "Kilobits"
          }
        },
        "MEGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits"
          }
        },
        "GIGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits"
          }
        },
        "TERABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits"
          }
        },
        "PERCENT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Percent"
          }
        },
        "BYTES_PER_SECOND": {
          "target":
"smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bytes/Second"
          }
```

```
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {

"smithy.api#enumValue": "Terabytes/Second"
          }
        },
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
```

```
                }
            },

        "TERABITS_PER_SECOND": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Terabits/Second"
                }
            },
            "COUNT_PER_SECOND": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Count/Second"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#EntitlementUsage": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Entitlement usage name.</p>",
                    "smithy.api#required": {}
                }
            },
            "ConsumedValue": {
                "target": "com.amazonaws.licensemanager#String",

            "traits": {
                    "smithy.api#documentation": "<p>Resource usage consumed.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxCount": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
                }
            },
            "Unit": {
                "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
                "traits": {
                    "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
                    "smithy.api#required": {}
                }
            }
```

        },
        "traits": {
          "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#EntitlementUsageList":
{
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#EntitlementUsage"
        }
      },
      "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
        }
      },
      "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {

```
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>",
            "smithy.api#required": {}

  }
        },
        "DryRun": {
          "target": "com.amazonaws.licensemanager#Boolean",
          "traits": {
            "smithy.api#default": false,
            "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },

        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#FailedDependencyException": {
      "type": "structure",
      "members": {
        "Message": {
```

```
          "target": "com.amazonaws.licensemanager#Message"
        },
        "ErrorCode": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "FailedDependency",
          "httpResponseCode": 424
        },
        "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 424
      }
    },
    "com.amazonaws.licensemanager#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#FilterName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
          }
        },
        "Values": {
          "target": "com.amazonaws.licensemanager#FilterValues",
          "traits": {
            "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n      describe operation. Filters can be used
to match a set of resources by specific criteria,\n      such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanager#FilterLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
```

```
      "aws.protocols#awsQueryError": {
        "code": "FilterLimitExceeded",
        "httpResponseCode": 400
      },
      "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#FilterList": {
    "type": "list",
     "member": {
      "target": "com.amazonaws.licensemanager#Filter"
    }
  },
  "com.amazonaws.licensemanager#FilterName": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValue": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValues": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#FilterValue",
      "traits": {
        "smithy.api#xmlName": "item"
      }
    }
  },
  "com.amazonaws.licensemanager#Filters": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Filter",
      "traits": {
        "smithy.api#xmlName": "item"
      }
    }
  },
  "com.amazonaws.licensemanager#GetAccessToken": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
    },
    "errors": [
```

```
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
 tokens\n        are valid for one hour.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenRequest": {
          "type": "structure",
          "members": {
            "Token": {
              "target": "com.amazonaws.licensemanager#TokenString",
              "traits": {
                "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
                "smithy.api#required": {}
              }
            },
            "TokenProperties": {
              "target": "com.amazonaws.licensemanager#MaxSize3StringList",
              "traits": {
                "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
          "type": "structure",
          "members": {
```

```
      "AccessToken": {
        "target": "com.amazonaws.licensemanager#TokenString",
        "traits": {
          "smithy.api#documentation": "<p>Temporary access token.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#GetGrant": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#GetGrantRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#GetGrantResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
  }
},
"com.amazonaws.licensemanager#GetGrantRequest": {
```

```
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
           "traits": {
             "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
             "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrantResponse": {
      "type": "structure",
      "members": {
        "Grant": {
          "target": "com.amazonaws.licensemanager#Grant",
          "traits": {
            "smithy.api#documentation": "<p>Grant details.</p>"
          }
        }
      },
      "traits": {

"smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#GetLicense": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetLicenseRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetLicenseResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
```

```
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }

      },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
```

```
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits":
{
          "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurationId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
          }
        },
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
```

```
        "traits": {
          "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",

        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License configuration status.</p>"
        }

      },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
        }
      },
      "ConsumedLicenseSummaryList": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
        }
      },
```

```
"ManagedResourceSummaryList": {
  "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
  "traits": {
    "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
  }
},
"Tags": {
  "target": "com.amazonaws.licensemanager#TagList",
  "traits": {
    "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
  }
},
"ProductInformationList": {
  "target": "com.amazonaws.licensemanager#ProductInformationList",
  "traits": {
    "smithy.api#documentation": "<p>Product information.</p>"
  }
},
"AutomatedDiscoveryInformation": {
  "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
  "traits": {
    "smithy.api#documentation": "<p>Automated discovery information.</p>"
  }
},
"DisassociateWhenNotFound": {
  "target": "com.amazonaws.licensemanager#BoxBoolean",
  "traits": {
    "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
  }
}
}
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanager#GetLicenseConversionTask": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
```

```
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {

          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
      "type": "structure",
      "members": {
        "LicenseConversionTaskId": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
          "traits": {
            "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
      "type": "structure",
      "members": {
        "LicenseConversionTaskId": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
          "traits": {
            "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
          }
        },
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
```

```
                }
            },
            "SourceLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
                }
            },
            "DestinationLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
                }
            },

    "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                }
            },
            "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                }
            }
        },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn": {
```

```
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
              "smithy.api#required":
{}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {
          "ReportGenerator": {
            "target": "com.amazonaws.licensemanager#ReportGenerator",
            "traits": {
              "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

```
            },
        "com.amazonaws.licensemanager#GetLicenseResponse": {
            "type": "structure",
            "members": {
                "License": {
                    "target": "com.amazonaws.licensemanager#License",
                    "traits": {
                        "smithy.api#documentation": "<p>License details.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },

"com.amazonaws.licensemanager#GetLicenseUsage": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {

"target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
    }
```

```
    },
    "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
      "type": "structure",
      "members": {
        "LicenseUsage": {
          "target":
"com.amazonaws.licensemanager#LicenseUsage",
          "traits": {
            "smithy.api#documentation": "<p>License usage details.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "S3BucketArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n   and so on.</p>"

  }
          },
          "SnsTopicArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
            }
          },
          "OrganizationConfiguration": {
            "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n   cross-account discovery.</p>"
            }
          },
          "EnableCrossAccountsDiscovery": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
            }

```
        },
          "LicenseManagerResourceShareArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n       provides member accounts with access to this share.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#Grant": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          },
          "GrantName": {

  "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant name.</p>",
              "smithy.api#required": {}
            }
          },
          "ParentArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Parent ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "GranteePrincipalArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The grantee principal ARN.</p>",
                    "smithy.api#required": {}
                }
            },
            "HomeRegion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantStatus": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>",
                    "smithy.api#required": {}
                }
            },
            "StatusReason": {
                "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status reason.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant version.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantedOperations": {
                "target": "com.amazonaws.licensemanager#AllowedOperationList",
                "traits": {
                    "smithy.api#documentation": "<p>Granted operations.</p>",
                    "smithy.api#required": {}
                }
            },
            "Options": {
                "target": "com.amazonaws.licensemanager#Options",
                "traits": {
                    "smithy.api#documentation": "<p>The options specified for the grant.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation":
```

"<p>Describes a grant.</p>"
            }
        },
        "com.amazonaws.licensemanager#GrantList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#Grant"
            }
        },
        "com.amazonaws.licensemanager#GrantStatus": {
            "type": "enum",
            "members": {
                "PENDING_WORKFLOW": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "PENDING_WORKFLOW"
                    }
                },
                "PENDING_ACCEPT": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "PENDING_ACCEPT"
                    }
                },
                "REJECTED": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "REJECTED"

                    }
                },
                "ACTIVE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "ACTIVE"
                    }
                },
                "FAILED_WORKFLOW": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "FAILED_WORKFLOW"
                    }
                },
                "DELETED": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "DELETED"
                    }

```
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits":
{

            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>"
          }

    },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
```

```
      }
    },
    "Issuer": {
      "target": "com.amazonaws.licensemanager#IssuerDetails",
      "traits": {
        "smithy.api#documentation": "<p>Granted license issuer.</p>"
      }
    },
    "HomeRegion": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanager#LicenseStatus",
      "traits": {
        "smithy.api#documentation": "<p>Granted license status.</p>"
      }
    },
    "Validity": {
      "target": "com.amazonaws.licensemanager#DatetimeRange",
      "traits": {
        "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
ISO8601-UTC format.</p>"
      }
    },
    "Beneficiary": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
      }
    },
    "Entitlements": {
      "target": "com.amazonaws.licensemanager#EntitlementList",
      "traits": {
        "smithy.api#documentation": "<p>License entitlements.</p>"
      }
    },
    "ConsumptionConfiguration": {
      "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
      "traits": {
        "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
      }
    },
    "LicenseMetadata": {
      "target": "com.amazonaws.licensemanager#MetadataList",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Granted license metadata.</p>"
          }
        },
        "CreateTime": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

    }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Version of the granted license.</p>"
          }
        },
        "ReceivedMetadata": {
          "target": "com.amazonaws.licensemanager#ReceivedMetadata",
          "traits": {
            "smithy.api#documentation": "<p>Granted license received metadata.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
      }
    },
    "com.amazonaws.licensemanager#GrantedLicenseList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#GrantedLicense"
      }
    },
    "com.amazonaws.licensemanager#ISO8601DateTime":
{
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
      }
    },
    "com.amazonaws.licensemanager#Integer": {
      "type": "integer"
```

```
        },
      "com.amazonaws.licensemanager#InvalidParameterValueException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidParameterValueProvided",
            "httpResponseCode": 400
          },
          "smithy.api#documentation":
  "<p>One or more parameter values are not valid.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#InvalidResourceStateException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidResourceState",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },

      "com.amazonaws.licensemanager#InventoryFilter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the filter.</p>",
              "smithy.api#required": {}
            }
```

```
        },
        "Condition": {
          "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
          "traits": {
            "smithy.api#documentation": "<p>Condition of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }

      },
        "traits": {
          "smithy.api#documentation": "<p>An inventory filter.</p>"
        }
      },
      "com.amazonaws.licensemanager#InventoryFilterCondition": {
        "type": "enum",
        "members": {
          "EQUALS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "EQUALS"
            }
          },
          "NOT_EQUALS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "NOT_EQUALS"
            }
          },
          "BEGINS_WITH": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "BEGINS_WITH"
            }
          },
          "CONTAINS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CONTAINS"
            }
          }
        }
```

```
        },
        "com.amazonaws.licensemanager#InventoryFilterList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#InventoryFilter"
            }
        },
        "com.amazonaws.licensemanager#Issuer": {
            "type": "structure",
            "members": {
                "Name": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Issuer name.</p>",
                        "smithy.api#required": {}
                    }
                },
                "SignKey": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
 of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
            }
        },
        "com.amazonaws.licensemanager#IssuerDetails": {
            "type": "structure",
            "members": {
                "Name": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Issuer name.</p>"
                    }
                },
                "SignKey": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
                    }
                },
                "KeyFingerprint": {
```

```
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
  }
},
"com.amazonaws.licensemanager#License": {
  "type": "structure",
  "members": {
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
      }
    },
    "LicenseName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {

    "smithy.api#documentation": "<p>License name.</p>"
      }
    },
    "ProductName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product name.</p>"
      }
    },
    "ProductSKU": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product SKU.</p>"
      }
    },
    "Issuer": {
      "target": "com.amazonaws.licensemanager#IssuerDetails",
      "traits": {
        "smithy.api#documentation": "<p>License issuer.</p>"
      }
    },
    "HomeRegion": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
```

```
                "smithy.api#documentation": "<p>Home Region of the license.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanager#LicenseStatus",
              "traits": {
                "smithy.api#documentation": "<p>License status.</p>"
              }
            },
            "Validity": {
              "target": "com.amazonaws.licensemanager#DatetimeRange",
              "traits": {
                "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
              }
            },
            "Beneficiary": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License beneficiary.</p>"
              }
            },
            "Entitlements": {

        "target": "com.amazonaws.licensemanager#EntitlementList",
              "traits": {
                "smithy.api#documentation": "<p>License entitlements.</p>"
              }
            },
            "ConsumptionConfiguration": {
              "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
              "traits": {
                "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
              }
            },
            "LicenseMetadata": {
              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                "smithy.api#documentation": "<p>License metadata.</p>"
              }
            },
            "CreateTime": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>License
creation time.</p>"
              }
            },
```

```
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
    }
  }
},
"com.amazonaws.licensemanager#LicenseConfiguration": {
  "type": "structure",
  "members": {
    "LicenseConfigurationId": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
      }
    },
    "LicenseConfigurationArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
      }
    },
    "Name": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Name of the license configuration.</p>"
      }
    },
    "Description": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Description of the license configuration.</p>"
      }
    },
    "LicenseCountingType": {
      "target": "com.amazonaws.licensemanager#LicenseCountingType",
      "traits": {
        "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
      }
    },
    "LicenseRules": {
      "target": "com.amazonaws.licensemanager#StringList",
      "traits": {
        "smithy.api#documentation": "<p>License rules.</p>"
```

```
                    }
                },
                "LicenseCount": {
                    "target": "com.amazonaws.licensemanager#BoxLong",
                    "traits": {
                        "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
                    }
                },
                "LicenseCountHardLimit": {
                    "target": "com.amazonaws.licensemanager#BoxBoolean",
                    "traits": {
                        "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
                    }
                },
                "DisassociateWhenNotFound": {
                    "target": "com.amazonaws.licensemanager#BoxBoolean",
                    "traits": {
                        "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                    }
                },
                "ConsumedLicenses": {
                    "target": "com.amazonaws.licensemanager#BoxLong",
                    "traits": {
                        "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Status of the license configuration.</p>"
                    }
                },
                "OwnerAccountId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

                    }
                },
                "ConsumedLicenseSummaryList": {
                    "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
                    "traits": {
                        "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
                    }
                },
                "ManagedResourceSummaryList": {
                    "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
```

```
        "traits": {
          "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
        }
      },
      "ProductInformationList": {
        "target": "com.amazonaws.licensemanager#ProductInformationList",
        "traits": {
          "smithy.api#documentation": "<p>Product information.</p>"
        }
      },
      "AutomatedDiscoveryInformation": {

        "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
        "traits": {
          "smithy.api#documentation": "<p>Automated discovery information.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n        consumed and enforced by License Manager. Components include specifications for
the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n        must be associated with a
host), and the number of licenses purchased and used.</p>"
    }
  },
  "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
    "type": "structure",
    "members": {
      "ResourceArn":

{
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Type of server resource.</p>"
        }
      },
      "ResourceOwnerId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
        }
```

```
        },
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
          }
        },
        "AmiAssociationScope": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
      "type": "enum",
      "members":
{
        "AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISABLED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
      "type": "structure",
      "members": {
        "ResourceArn": {
```

```
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits":
{
            "smithy.api#documentation": "<p>Type of resource.</p>"
          }
        },
        "ResourceStatus": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the resource.</p>"
          }
        },
        "ResourceOwnerId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
          }
        },
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
          }
        },

    "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
```

```
          }
        },
      "com.amazonaws.licensemanager#LicenseConfigurations": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfiguration"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
        "members": {
          "UsageOperation": {
            "target": "com.amazonaws.licensemanager#UsageOperation",
            "traits": {
              "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n        </p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {

    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
```

```
          }
        },
        "DestinationLicenseContext": {
          "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the conversion task.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
          }
        },
        "StartTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation":
 "<p>The time the conversion task was started at.</p>"
          }
        },
        "LicenseConversionTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
          }
        },
        "EndTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskId": {
```

```
        "type": "string",
        "traits": {

      "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
      "type": "enum",
      "members": {
        "IN_PROGRESS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "IN_PROGRESS"
          }
        },
        "SUCCEEDED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUCCEEDED"
          }
        },
        "FAILED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTasks":
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConversionTask"
      }
    },
    "com.amazonaws.licensemanager#LicenseCountingType": {
      "type": "enum",
      "members": {
        "VCPU": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "vCPU"
          }
```

```
        },
        "INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Instance"
          }
        },
        "CORE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Core"
          }
        },
        "SOCKET": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Socket"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseDeletionStatus": {
      "type": "enum",
      "members": {
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#License"
      }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailure": {

      "type": "structure",
      "members": {
```

```
"ResourceArn": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
  }
},
"ResourceType": {
  "target": "com.amazonaws.licensemanager#ResourceType",
  "traits": {
    "smithy.api#documentation": "<p>Resource type.</p>"
  }
},
"ErrorMessage": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>Error message.</p>"
  }
},
"FailureTime": {
  "target": "com.amazonaws.licensemanager#DateTime",
  "traits": {

    "smithy.api#documentation": "<p>Failure time.</p>"
  }
},
"OperationName": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>Name of the operation.</p>"
  }
},
"ResourceOwnerId": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
  }
},
"OperationRequestedBy": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
  }
},
"MetadataList": {

  "target": "com.amazonaws.licensemanager#MetadataList",
  "traits": {
```

```
            "smithy.api#documentation": "<p>Reserved.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailureList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
      }
    },
    "com.amazonaws.licensemanager#LicenseSpecification": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

            "smithy.api#required": {}
          }
        },
        "AmiAssociationScope": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseSpecifications": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseSpecification"
      }
    },
    "com.amazonaws.licensemanager#LicenseStatus": {
      "type": "enum",
      "members": {
        "AVAILABLE": {
```

```
          "target": "smithy.api#Unit",

        "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "PENDING_AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_AVAILABLE"
          }
        },
        "DEACTIVATED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DEACTIVATED"
          }
        },
        "SUSPENDED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUSPENDED"
          }
        },
        "EXPIRED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EXPIRED"
          }
        },

      "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseUsage": {
      "type": "structure",
      "members": {
```

```
    "EntitlementUsages": {
      "target": "com.amazonaws.licensemanager#EntitlementUsageList",
      "traits": {
        "smithy.api#documentation": "<p>License entitlement usages.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"


  }
},
"com.amazonaws.licensemanager#LicenseUsageException": {
  "type": "structure",
  "members": {
    "Message": {
      "target": "com.amazonaws.licensemanager#Message"
    }
  },
  "traits": {
    "aws.protocols#awsQueryError": {
      "code": "LicenseUsageFailure",
      "httpResponseCode": 412
    },
    "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 412
  }
},
"com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
  },
  "output": {

"target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n          <p>Resource associations need not consume licenses from a license
configuration. \n          For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
          }
        },
        "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
          "type": "structure",
```

```
      "members": {
        "LicenseConfigurationAssociations": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
          "traits": {
            "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
          }
        },
        "NextToken": {

 "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrants": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
```

              "target": "com.amazonaws.licensemanager#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
         }
      },
      "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
         "type": "structure",
         "members": {
            "GrantArns": {
               "target": "com.amazonaws.licensemanager#ArnList",
               "traits": {

               "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanager#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>LicenseArn</code>\n   </p>\n
      \n   \n   <p>\n   <code>GrantStatus</code>\n   </p>\n   \n
  \n   <p>\n   <code>GranteePrincipalARN</code>\n   </p>\n   \n
\n   <p>\n   <code>ProductSKU</code>\n   </p>\n   \n   \n
<p>\n   <code>LicenseIssuerName</code>\n
      </p>\n   \n   "
               }
            },
            "NextToken": {
               "target": "com.amazonaws.licensemanager#String",
               "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanager#MaxSize100",
               "traits": {
                  "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
         "type": "structure",

```
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",

    "traits": {
              "smithy.api#documentation": "<p>Distributed grant details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
      },
      "errors": [
        {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
      }
```

        },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
             "traits": {
               "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
               "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
        "type": "structure",
        "members": {
          "LicenseOperationFailureList": {
            "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
            "traits": {
              "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }

        },
      "com.amazonaws.licensemanager#ListLicenseConfigurations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
        },

     "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configurations
for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {

```
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanager#Filters",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n            <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n            Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n            Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n
<p>\n            <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n        Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
    </li>\n        <li>\n            <p>\n            <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
        "type": "structure",
        "members": {
            "LicenseConfigurations": {
                "target": "com.amazonaws.licensemanager#LicenseConfigurations",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license configurations.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",

            "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
      "type": "structure",
      "members": {
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {

"smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
```

```
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#Filters",
          "traits": {
            "smithy.api#documentation": "<p>\n        Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n      </p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
      "type": "structure",
      "members": {
        "LicenseConversionTasks": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target":
```

"com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n   \n   \n   <p>\n   <code>LicenseConfigurationArn</code>\n
</p>\n   \n   "
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits":
{
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
      "type": "structure",
      "members": {
        "ReportGenerators": {
          "target": "com.amazonaws.licensemanager#ReportGeneratorList",
          "traits": {
            "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }

        }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
```

```
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
          "type": "structure",
          "members": {
            "LicenseSpecifications": {
              "target": "com.amazonaws.licensemanager#LicenseSpecifications",
              "traits": {
                "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
              }
            },
```

```
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#ListLicenseVersions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
  }
},
"com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
  "type": "structure",
  "members": {
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
        "smithy.api#required": {}
```

```
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                     "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
            "type": "structure",
            "members": {
                "Licenses": {
                    "target": "com.amazonaws.licensemanager#LicenseList",
                    "traits": {
                        "smithy.api#documentation": "<p>License details.</p>"
                    }
                },

"NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListLicenses": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListLicensesRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#ListLicensesResponse"
            },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },

    "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n        <p>\n            <code>Beneficiary</code>\n        </p>\n
    </li>\n        <li>\n        <p>\n            <code>ProductSKU</code>\n        </p>\n        </li>\n
  <li>\n        <p>\n            <code>Fingerprint</code>\n        </p>\n        </li>\n        <li>\n
<p>\n            <code>Status</code>\n        </p>\n        </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token
```

for the next set of results.</p>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                  "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#ListLicensesResponse": {
            "type": "structure",
            "members": {
              "Licenses": {
                "target": "com.amazonaws.licensemanager#LicenseList",
                "traits": {
                  "smithy.api#documentation": "<p>License details.</p>"
                }
              },
              "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {

      "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#ListReceivedGrants": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {

```
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
             },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n        recipient as this Amazon Web Services account, your organization, or an organizational unit\n
    (OU) to which this member account belongs.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
```

```
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target":
"com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
              "smithy.api#required": {}
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
    <ul>\n         <li>\n          <p>\n            <code>ParentArn</code>\n         </p>\n         </li>\n
<li>\n          <p>\n            <code>GranteePrincipalArn</code>\n         </p>\n         </li>\n       </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

"com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
    "type": "structure",
    "members": {
        "Grants": {
            "target": "com.amazonaws.licensemanager#GrantList",
            "traits": {
                "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
    "type": "structure",
    "members": {
        "GrantArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>ProductSKU</code>\n   </p>\n
    \n   \n   <p>\n   <code>LicenseIssuerName</code>\n   </p>\n
\n   \n   <p>\n   <code>LicenseArn</code>\n   </p>\n   \n
\n   <p>\n   <code>GrantStatus</code>\n   </p>\n   \n   \n
<p>\n
    <code>GranterAccountId</code>\n   </p>\n   \n   "
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        },

```
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#MaxSize100",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
},
"com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
  "type": "structure",
  "members": {
    "Grants": {
      "target": "com.amazonaws.licensemanager#GrantList",
      "traits": {
        "smithy.api#documentation": "<p>Received grant details.</p>"
      }
    },
    "NextToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#ListReceivedLicenses": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```json
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists received licenses.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {

          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
```

```
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanager#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>Beneficiary</code>\n                </p>\n
</li>\n            <li>\n                <p>\n                    <code>ProductSKU</code>\n
                </p>\n            </li>\n        </ul>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
          "type": "structure",
          "members": {
            "Licenses": {
              "target": "com.amazonaws.licensemanager#GrantedLicenseList",
              "traits": {
                "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
```

```
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
        "type": "structure",
        "members": {
          "LicenseArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
            }
          },

          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n            <li>\n              <p>\n                  <code>ProductSKU</code>\n              </p>\n
          </li>\n          <li>\n              <p>\n                  <code>Status</code>\n              </p>\n          </li>\n
<li>\n          <p>\n                  <code>Fingerprint</code>\n              </p>\n          </li>\n          <li>\n
<p>\n              <code>IssuerName</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
<code>Beneficiary</code>\n              </p>\n          </li>\n          </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
        "type": "structure",
```

```
    "members": {
      "Licenses": {
        "target": "com.amazonaws.licensemanager#GrantedLicenseList",
        "traits": {
          "smithy.api#documentation": "<p>Received license details.</p>"
        }
      },

    "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#ListResourceInventory": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
       "target": "com.amazonaws.licensemanager#FailedDependencyException"
    },
    {
      "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#InventoryFilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n          are supported:</p>\n          <ul>\n            <li>\n              <p>\n                  <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n              Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
            </li>\n            <li>\n              <p>\n                  <code>application_name</code> - The name of the
application. \n              Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
          </li>\n            <li>\n              <p>\n                  <code>license_included</code> - The type of license included. \n
          Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n              Possible values
are <code>sql-server-enterprise</code> | \n              <code>sql-server-standard</code> | \n              <code>sql-
server-web</code> | \n              <code>windows-server-datacenter</code>.</p>\n            </li>\n            <li>\n
      <p>\n                  <code>platform</code> - The platform of the resource.\n              Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n            </li>\n            <li>\n              <p>\n
          <code>resource_id</code> - The ID of the resource. \n              Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n            </li>\n            <li>\n              <p>\n
    <code>tag:<key></code> - The key/value combination of a tag assigned \n              to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n              <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n            </li>\n          </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

```json
      "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
        "type": "structure",
        "members": {
          "ResourceInventoryList": {
            "target": "com.amazonaws.licensemanager#ResourceInventoryList",
            "traits": {
              "smithy.api#documentation": "<p>Information
about the resources.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
```

```
                }
            },
        "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
            "type": "structure",
            "members": {
                "ResourceArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the tags.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListTokens": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListTokensRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListTokensResponse"
        },
        "errors":
[
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists your tokens.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTokensRequest": {
      "type": "structure",
      "members": {
        "TokenIds": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {

            "smithy.api#documentation": "<p>Token IDs.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
  <ul>\n          <li>\n            <p>\n              <code>LicenseArns</code>\n            </p>\n          </li>\n
</ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
```

```
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
            "Tokens": {
                "target": "com.amazonaws.licensemanager#TokenList",
                "traits": {
                    "smithy.api#documentation": "<p>Received token details.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n        consumption details by resource at a selected point in time. Use this action to audit the\n        current
license consumption for any license inventory and configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
    "type": "structure",
    "members": {
      "LicenseConfigurationArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#Filters",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n
<li>\n        <p>\n        <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n        <p>\n
    <code>resourceAccount</code> - The ID of the account that owns the resource. \n        Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        </ul>"
        }
      }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {

"LicenseConfigurationUsageList": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      }
    },
    "com.amazonaws.licensemanager#Location": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Long": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#ManagedResourceSummary": {
      "type": "structure",
      "members": {
        "ResourceType":
{
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
            }
          },
          "AssociationCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
            }
          }
```

```
      },
      "traits": {
        "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
      }
    },
    "com.amazonaws.licensemanager#MaxSize100": {

      "type": "integer",
      "traits": {
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "com.amazonaws.licensemanager#MaxSize3StringList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 3
        }
      }
    },
    "com.amazonaws.licensemanager#Message": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Metadata": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The key name.</p>"
          }
        },

    "Value": {
          "target": "com.amazonaws.licensemanager#String",
```

```
              "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
          }
        },
        "com.amazonaws.licensemanager#MetadataList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Metadata"
          }
        },
        "com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#Options": {
          "type": "structure",
          "members": {
            "ActivationOverrideBehavior": {
              "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
              "traits": {
                "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n        Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n        <ul>\n          <li>\n            <p>As a license administrator, you can
optionally specify an\n            <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
  </li>\n          <li>\n            <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n            account in your
organization.</p>\n          </li>\n          <li>\n            <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n            <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n          are activating the grant.</p>\n          </li>\n        </ul>\n        <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n          <dd>\n            <p>Use this value to activate a grant
without replacing any member accounts\n            active grants for the same product.</p>\n          </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n          <dd>\n            <p>Use this value to activate a
```

grant and disable other active grants in any\n   member accounts
 for the same product. This action will also replace their\n   previously activated grants with this activated
grant.</p>\n   </dd>\n   </dl>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n   override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "com.amazonaws.licensemanager#OrganizationConfiguration": {
            "type": "structure",
            "members": {
                "EnableIntegration": {
                    "target": "com.amazonaws.licensemanager#Boolean",
                    "traits": {
                        "smithy.api#default": false,
                        "smithy.api#documentation":
"<p>Enables Organizations integration.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
            }
        },
        "com.amazonaws.licensemanager#PrincipalArnList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#Arn"
            },
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 1
                }
            }
        },
        "com.amazonaws.licensemanager#ProductInformation": {
            "type": "structure",
            "members": {
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {

```
                "smithy.api#documentation": "<p>Resource type. The
 possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                "smithy.api#required": {}
              }
            },
          "ProductInformationFilterList": {
            "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
            "traits": {
              "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n        <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n        <p>The following filters and are supported when
the resource type \n        is <code>SSM_MANAGED</code>:</p>\n
        <ul>\n          <li>\n            <p>\n              <code>Application Name</code> - The name of the
application.\n          Logical operator is <code>EQUALS</code>.</p>\n          </li>\n          <li>\n
<p>\n              <code>Application Publisher</code> - The publisher of the application.\n          Logical
operator is <code>EQUALS</code>.</p>\n          </li>\n          <li>\n          <p>\n
<code>Application Version</code> - The version of the application.\n          Logical operator is
<code>EQUALS</code>.</p>\n          </li>\n          <li>\n          <p>\n              <code>Platform
Name</code> - The name of the platform.\n          Logical operator is <code>EQUALS</code>.</p>\n
</li>\n          <li>\n          <p>\n              <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
         </li>\n          <li>\n          <p>\n              <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>.  The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n          </p>\n          </li>\n          <li>\n          <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n          Logical operator is <code>NOT_EQUALS</code>.</p>\n          </li>\n
<li>\n          <p>\n              <code>License Included</code> - The type of license included.\n          Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n          Possible values are:
 <code>sql-server-enterprise</code> | \n          <code>sql-server-standard</code> | \n          <code>sql-server-
web</code> | \n          <code>windows-server-datacenter</code>.</p>\n          </li>\n          </ul>\n
<p>The following filters and logical operators are supported when the resource type\n          is
<code>RDS</code>:</p>\n        <ul>\n          <li>\n          <p>\n              <code>Engine Edition</code> -
The edition of the database engine.\n          Logical operator is <code>EQUALS</code>.\n          Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n          </li>\n          <li>\n          <p>\n              <code>License Pack</code> - The license
pack.\n          Logical operator is <code>EQUALS</code>.\n          Possible values are: <code>data
guard</code> | \n          <code>diagnostic pack sqlt</code> | \n
        <code>tuning pack sqlt</code> | \n          <code>ols</code> | \n          <code>olap</code>.</p>\n
   </li>\n          </ul>",
                "smithy.api#required": {}
              }
            }
          },
```

```
    "traits": {
      "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
    }
},
"com.amazonaws.licensemanager#ProductInformationFilter": {
  "type": "structure",
  "members": {
    "ProductInformationFilterName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Filter name.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductInformationFilterValue": {
      "target": "com.amazonaws.licensemanager#StringList",

      "traits": {
        "smithy.api#documentation": "<p>Filter value.</p>"
      }
    },
    "ProductInformationFilterComparator": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Logical operator.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes product information filters.</p>"
  }
},
"com.amazonaws.licensemanager#ProductInformationFilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanager#ProductInformationFilter"
  }
},
"com.amazonaws.licensemanager#ProductInformationList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanager#ProductInformation"
  }
},
"com.amazonaws.licensemanager#ProvisionalConfiguration": {
  "type": "structure",
  "members": {
```

```
      "MaxTimeToLiveInMinutes": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#RateLimitExceededException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "RateLimitExceeded",
        "httpResponseCode": 429
      },
      "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 429
    }
  },
  "com.amazonaws.licensemanager#ReceivedMetadata": {
    "type": "structure",
    "members": {
      "ReceivedStatus": {
        "target": "com.amazonaws.licensemanager#ReceivedStatus",
        "traits": {
          "smithy.api#documentation": "<p>Received status.</p>"
        }
      },
      "ReceivedStatusReason": {
        "target": "com.amazonaws.licensemanager#StatusReasonMessage",
        "traits": {
          "smithy.api#documentation": "<p>Received
status reason.</p>"
        }
      },
      "AllowedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
```

```
              "traits": {
                "smithy.api#documentation": "<p>Allowed operations.</p>"
              }
            }
          },
        "traits": {
          "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
        }
      },
    "com.amazonaws.licensemanager#ReceivedStatus": {
      "type": "enum",
      "members": {
        "PENDING_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_WORKFLOW"
          }
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"PENDING_ACCEPT"
          }
        },
        "REJECTED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"
          }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
```

```
      },
      "DISABLED": {
        "target": "smithy.api#Unit",
         "traits": {
           "smithy.api#enumValue": "DISABLED"
         }
      },
      "WORKFLOW_COMPLETED": {
        "target": "smithy.api#Unit",
        "traits": {
           "smithy.api#enumValue": "WORKFLOW_COMPLETED"
         }
      }
    }
  }
},
"com.amazonaws.licensemanager#RedirectException": {
  "type": "structure",
  "members": {
    "Location": {
      "target": "com.amazonaws.licensemanager#Location",
      "traits": {
         "smithy.api#httpHeader": "Location"
       }
    },
    "Message": {
      "target": "com.amazonaws.licensemanager#Message"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
     "smithy.api#error": "client",
    "smithy.api#httpError": 308
  }
},
"com.amazonaws.licensemanager#RejectGrant": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#RejectGrantRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#RejectGrantResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#RejectGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#RejectGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
```

```json
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#RenewType":
{
      "type": "enum",
      "members": {
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "WEEKLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Weekly"
          }
        },
        "MONTHLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Monthly"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportContext": {
      "type": "structure",
      "members": {
        "licenseConfigurationArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
```

```
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportFrequency": {
        "type": "structure",
        "members": {
            "value": {
                "target": "com.amazonaws.licensemanager#Integer",
                "traits": {
                    "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
                }
            },
            "period": {
                "target": "com.amazonaws.licensemanager#ReportFrequencyType",

                "traits": {
                    "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportFrequencyType": {
        "type": "enum",
        "members": {
            "DAY": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DAY"
                }
            },
            "WEEK": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "WEEK"
                }
            },
```

```
            "MONTH": {
              "target": "smithy.api#Unit",
              "traits":
{
                "smithy.api#enumValue": "MONTH"
              }
            }
          }
        }
      },
      "com.amazonaws.licensemanager#ReportGenerator": {
        "type": "structure",
        "members": {
          "ReportGeneratorName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the report generator.</p>"
            }
          },
          "ReportType": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
              "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
            }
          },
          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
            }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits": {
              "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
            }
          },
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
            }
          },
          "LastRunStatus": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
            }
```

```
        },
        "LastRunFailureReason":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
            }
        },
        "LastReportGenerationTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
            }
        },
        "ReportCreatorAccount": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
            }
        },
        "Description": {
            "target": "com.amazonaws.licensemanager#String",

        "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
        },
        "S3Location": {
            "target": "com.amazonaws.licensemanager#S3Location",
            "traits": {
                "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
            }
        },
        "CreateTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Time the report was created.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
                "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
            }
```

```
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
    }
  },
  "com.amazonaws.licensemanager#ReportGeneratorList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ReportGenerator"
    }
  },
  "com.amazonaws.licensemanager#ReportGeneratorName": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "com.amazonaws.licensemanager#ReportType": {
    "type": "enum",
    "members": {
      "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
        }

      },
      "LICENSE_CONFIGURATION_USAGE_REPORT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "LicenseConfigurationUsageReport"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#ReportTypeList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ReportType"
    }
  },
  "com.amazonaws.licensemanager#ResourceInventory": {
    "type": "structure",
    "members": {
```

```
      "ResourceId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits":

{

          "smithy.api#documentation": "<p>Type of resource.</p>"
        }
      },
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "Platform": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Platform of the resource.</p>"
        }
      },
      "PlatformVersion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
        }
      },
      "ResourceOwningAccountId":

{

        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about a resource.</p>"
    }
  },
  "com.amazonaws.licensemanager#ResourceInventoryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ResourceInventory"
    }
```

```
        },
        "com.amazonaws.licensemanager#ResourceLimitExceededException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "ResourceLimitExceeded",

              "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#ResourceNotFoundException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "InvalidResource.NotFound",
              "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>The resource cannot be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#ResourceType": {
          "type": "enum",
          "members": {
            "EC2_INSTANCE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "EC2_INSTANCE"
              }
            },
            "EC2_HOST": {
              "target": "smithy.api#Unit",
              "traits": {
```

```
            "smithy.api#enumValue": "EC2_HOST"
          }
        },
        "EC2_AMI": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_AMI"
          }
        },
        "RDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "RDS"
          }
        },
        "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#S3Location": {
      "type": "structure",
      "members": {
        "bucket": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
          }
        },
        "keyPrefix": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
      }
    },
    "com.amazonaws.licensemanager#ServerInternalException": {
      "type": "structure",
      "members": {
        "Message": {
```

```
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "InternalError",
        "httpResponseCode": 500
      },
      "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanager#SignedToken": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 4096
      }
    }
  },
  "com.amazonaws.licensemanager#StatusReasonMessage": {
    "type": "string",
    "traits":
{
      "smithy.api#length": {
        "min": 0,
        "max": 400
      },
      "smithy.api#pattern": "^[\\s\\S]+$"
    }
  },
  "com.amazonaws.licensemanager#String": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#StringList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#String"
    }
  },
  "com.amazonaws.licensemanager#Tag": {
    "type": "structure",
    "members": {
      "Key": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Tag key.</p>"
```

```
              }
            },
            "Value": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation":
"<p>Tag value.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#TagKeyList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#String"
          }
        },
        "com.amazonaws.licensemanager#TagList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Tag"
          }
        },
        "com.amazonaws.licensemanager#TagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#TagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#TagResourceResponse"
          },
          "errors": [
            {
              "target":
"com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#TagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits":
{
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>One or more tags.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#TokenData": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
```

```
    "TokenType": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
      }
    },
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
      }
    },
    "ExpirationTime": {
      "target": "com.amazonaws.licensemanager#ISO8601DateTime",
      "traits": {
        "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
 format.</p>"
      }
    },
    "TokenProperties": {
      "target": "com.amazonaws.licensemanager#MaxSize3StringList",
      "traits": {
        "smithy.api#documentation": "<p>Data specified by the caller.</p>"
      }
    },
    "RoleArns": {
      "target": "com.amazonaws.licensemanager#ArnList",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
      }
    }
  },
  "traits": {

    "smithy.api#documentation": "<p>Describes a token.</p>"
  }
},
"com.amazonaws.licensemanager#TokenList": {
  "type": "list",
```

```
      "member": {
        "target": "com.amazonaws.licensemanager#TokenData"
      }
    },
    "com.amazonaws.licensemanager#TokenString": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 4096
        },
        "smithy.api#pattern": "^\\S+$"
      }
    },
    "com.amazonaws.licensemanager#TokenType": {
      "type": "enum",
      "members": {
        "REFRESH_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REFRESH_TOKEN"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
{
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UntagResourceResponse"
      },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
         },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanager#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UntagResourceResponse": {
      "type": "structure",
      "members": {},
```

```
        "traits": {
          "smithy.api#output": {}
        }
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
      },
      "output": {

"target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
```

```
"LicenseConfigurationStatus": {
    "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
    "traits": {
        "smithy.api#documentation": "<p>New status of the license configuration.</p>"
    }
},
"LicenseRules": {
    "target": "com.amazonaws.licensemanager#StringList",
    "traits": {

        "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
        configuration is licenseAffinityToHost.</p>"
    }
},
"LicenseCount": {
    "target": "com.amazonaws.licensemanager#BoxLong",
    "traits": {
        "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
    }
},
"LicenseCountHardLimit": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
    "traits": {
        "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
    }
},
"Name": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
    }
},
"Description": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>New description of the license configuration.</p>"
    }
},
"ProductInformationList": {
    "target": "com.amazonaws.licensemanager#ProductInformationList",
    "traits": {
        "smithy.api#documentation": "<p>New product information.</p>"
    }
},
"DisassociateWhenNotFound": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
```

```
            "traits": {
                "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
            }
        }
    },
    "traits":
{
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
```

```
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportGeneratorName": {
            "target": "com.amazonaws.licensemanager#ReportGeneratorName",
            "traits": {
              "smithy.api#documentation": "<p>Name of the report generator.</p>",
              "smithy.api#required": {}
            }
          },
          "Type": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
              "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n        <ul>\n          <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n          </li>\n          <li>\n            <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n          </li>\n          </ul>",
              "smithy.api#required": {}
            }
          },
          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>The report context.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
```

```
                    "traits":
{
                        "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ClientToken": {
                    "target": "com.amazonaws.licensemanager#ClientRequestToken",
                    "traits": {
                        "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Description": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Description of the report generator.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
```

```
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },


        {
            "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
        },
        {
            "target": "com.amazonaws.licensemanager#LicenseUsageException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
    }
},

    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "AddLicenseSpecifications": {
                "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                "traits": {
                    "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
                }
            },
            "RemoveLicenseSpecifications": {
                "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                "traits": {
                    "smithy.api#documentation":
"<p>ARNs of the license configurations to remove.</p>"
                }
```

```
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
                }
            },
            "SnsTopicArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
                }
            },
            "OrganizationConfiguration": {
                "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
                }
            },
            "EnableCrossAccountsDiscovery": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

        }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UsageOperation": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            }
        }
    },
    "com.amazonaws.licensemanager#ValidationException": {
```

```
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

      "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {
```

```go
	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// Current version
	of the license.
	//
	// This member is required.
	SourceVersion *string

	noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

	// Date when the license is deleted.
	DeletionDate *string

	// License status.
	Status types.LicenseDeletionStatus

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack,
	options); err != nil {
		return err
```

```
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
	return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
	return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if
	err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
	return err
	}
	if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
	return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
	middleware.Before); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
	return err
	}
```

```
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
    Region:        region,
    ServiceID:     ServiceID,
    OperationName: "DeleteLicense",
    }
}
```

1.6 gogo-protobuf 1.3.2

1.6.1 Available under license :

Copyright (c) 2013, The GoGo Authors. All rights reserved.

Protocol Buffers for Go with Gadgets

Go support for Protocol Buffers - Google's data interchange format

Copyright 2010 The Go Authors. All rights reserved.
https://github.com/golang/protobuf

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Anton Povarov <anton.povarov@gmail.com>

Brian Goff <cpuguy83@gmail.com>

Clayton Coleman <ccoleman@redhat.com>

Denis Smirnov <denis.smirnov.91@gmail.com>

DongYun Kang <ceram1000@gmail.com>

Dwayne Schultz <dschultz@pivotal.io>

Georg Apitz <gapitz@pivotal.io>

Gustav Paul <gustav.paul@gmail.com>

Johan Brandhorst <johan.brandhorst@gmail.com>

John Shahid <jvshahid@gmail.com>

John Tuley <john@tuley.org>

Laurent <laurent@adyoulike.com>

Patrick Lee <patrick@dropbox.com>

Peter Edge <peter.edge@gmail.com>

Roger Johansson <rogeralsing@gmail.com>

Sam Nguyen <sam.nguyen@sendgrid.com>

Sergio Arbeo <serabe@gmail.com>

Stephen J Day <stephen.day@docker.com>

Tamir Duberstein <tamird@gmail.com>

Todd Eisenberger <teisenberger@dropbox.com>

Tormod Erevik Lea <tormodlea@gmail.com>

Vyacheslav Kim <kane@sendgrid.com>

Walter Schulze <awalterschulze@gmail.com>

1.7 aws-aws-lambda-go 1.41.0

1.7.1 Available under license :

Copyright 2017 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Lambda functions are made available under a modified MIT license.
See LICENSE-LAMBDACODE for details.

The remainder of the project is made available under the terms of the

Apache License, version 2.0. See LICENSE for details.
MIT No Attribution

Permission is hereby granted, free of charge, to any person obtaining a copy of this
software and associated documentation files (the "Software"), to deal in the Software
without restriction, including without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

  4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding
those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.8 protobuf 1.31.0

1.8.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

 * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above

copyright notice, this list of conditions and the following disclaimer

in the documentation and/or other materials provided with the

distribution.

   * Neither the name of Google Inc. nor the names of its

contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

OWNER

 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.9 go-uber-org-multierr 1.11.0

1.9.1 Available under license :

Copyright (c) 2017-2021 Uber Technologies, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION

 WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

1.10 genproto-googleapis-rpc 0.0.0-20230822172742-b8732ec3820d

1.10.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

 4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding
those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions

   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

 5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.11 zap 1.26.0

1.11.1 Available under license :

Copyright (c) 2016-2017 Uber Technologies, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.12 grpc-go 1.59.0

1.12.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
 incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
Copyright 2014 gRPC authors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.13 goprotobuf 1.5.3
1.13.1 Available under license :
This source code was written by the Go contributors.
The master list of contributors is in the main Go distribution,
visible at http://tip.golang.org/CONTRIBUTORS.
Copyright 2010 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.14 x-sys 0.14.0

1.14.1 Available under license :

Copyright (c) 2013 unformatt

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH
 THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.15 google-uuid 1.4.0

1.15.1 Available under license :

Paul Borman <borman@google.com>
bmatsuo
shawnps
theory
jboverfelt
dsymonds
cd1
wallclockbuilder
dansouza
Copyright (c) 2009,2014 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.16 x-text 0.14.0

1.16.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.17 x-net 0.18.0

1.17.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer

in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.18 x-sync 0.5.0

1.18.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

1.19 aws-aws-sdk-go-v2 1.25.1

1.19.1 Available under license :

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil,
 err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
```

```go
	//
	// This member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to associate the license with.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

	// A structure containing data about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
	if err !=
	nil {
		return err
	}
	err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
```

```
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack);
err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
```

```
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack,
 options); err != nil {
  return err
 }
 return nil
}


func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

```
name: License Scan

on: [pull_request]

jobs:
 licensescan:
  name: License Scan
  runs-on: ubuntu-latest
  strategy:
   matrix:
    python-version: [3.9]

  steps:
   - name: Checkout target
     uses: actions/checkout@v2
     with:
      path: sdkbase
      ref: ${{ github.base_ref }}
   - name: Checkout this ref
     uses: actions/checkout@v2
     with:
      path: new-ref
      fetch-depth: 0
   - name: Get Diff
     run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
```

```
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)


// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```go
	c.addOperationCheckInLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*CheckInLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type CheckInLicenseInput struct {

	// License consumption token.
	//
	// This member is required.
	LicenseConsumptionToken *string

	// License beneficiary.
	Beneficiary
	*string

	noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return
err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
```

```
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckInLicense",
 }
}
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
          }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
```

```
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n      Organizations integration has been enabled, and
which Regions data will be aggregated from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays
the aggregated data for Linux\n      subscriptions.</p>"
            }
          }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
```

```
"members": {
  "AmiId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
    }
  },
  "InstanceID": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
    }
  },
  "InstanceType": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The instance type of the resource.</p>"
    }
  }
},
  "AccountID": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
```

```
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
              "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
              }
            },
            "LastUpdatedTime": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
              }
            },
            "SubscriptionName": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
          "type": "service",
          "version": "2018-05-10",
          "operations": [
```

```
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],

  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
```

```
          "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The
 AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "Boolean"
              },
              "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

              "type": "Boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "String"
              }
            },
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Endpoint"
                      }
                    ]
                  }
```

```
        ],
        "type": "tree",
        "rules": [
            {

        "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
            "type": "error"
        },
        {
            "conditions": [],
            "type": "tree",
            "rules": [
                {

            "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],

        "endpoint": {
                    "url": {
                        "ref": "Endpoint"
                    },
                    "properties": {},
                    "headers": {}
```

```
                    },
                    "type": "endpoint"
                }
            ]
        }
    ]
},
{
    "conditions": [],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",

        "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],

        "assign": "PartitionResult"
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },
```

```
                              true
                           ]
                        },

                  {
                        "fn": "booleanEquals",
                        "argv": [
                           {
                              "ref": "UseDualStack"
                           },
                           true
                        ]
                     }
                  ],
                  "type": "tree",
                  "rules": [
                     {
                        "conditions": [
                           {

                     "fn": "booleanEquals",
                           "argv": [
                              true,
                              {
                                 "fn": "getAttr",
                                 "argv": [
                                    {
                                       "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                 ]

}

                           ]
                        },
                        {
                           "fn": "booleanEquals",
                           "argv": [
                              true,
                              {
                                 "fn": "getAttr",
                                 "argv": [
                                    {
                                       "ref": "PartitionResult"
                                    },

                           "supportsDualStack"
                                 ]
```

```
                                                }
                                            ]
                                        }
                                    ],
                                    "type": "tree",
                                    "rules": [
                                        {
                                            "conditions": [],
                                            "type": "tree",
                                            "rules": [
                                                {

                                                    "conditions": [],
                                                    "endpoint": {
                                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                        "properties": {},
                                                        "headers": {}
                                                    },
                                                    "type": "endpoint"
                                                }
                                            ]
                                        }
                                    ]

                                },
                                {
                                    "conditions": [],
                                    "error": "FIPS and DualStack are enabled, but this partition does not support

one or both",

                                    "type": "error"
                                }
                            ]
                        },
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {

                                            "ref": "UseFIPS"
                                        },
                                        true
                                    ]
                                }
                            ],
                            "type": "tree",
```

```
                                    "rules": [
                                      {
                                        "conditions": [
                                          {
                                            "fn": "booleanEquals",
                                            "argv": [
                                              true,

        {
                                                "fn": "getAttr",
                                                "argv": [
                                                  {
                                                    "ref": "PartitionResult"
                                                  },
                                                  "supportsFIPS"
                                                ]
                                              }
                                            ]
                                          }
                                        ],
                                        "type": "tree",

                        "rules": [
                                          {
                                            "conditions": [],
                                            "type": "tree",
                                            "rules": [
                                              {
                                                "conditions": [],
                                                "endpoint": {
                                                  "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                                  "properties": {},
                                                  "headers": {}

                                                },
                                                "type": "endpoint"
                                              }
                                            ]
                                          }
                                        ]
                                      },
                                      {
                                        "conditions": [],
                                        "error": "FIPS is enabled but this partition does not support FIPS",
                                        "type": "error"
                                      }
                                    ]
```

```
        },
                            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {

                     "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {
                                "conditions": [],
                                "type": "tree",
                                "rules": [
                                    {
                                        "conditions": [],

                                 "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
```

```
                                                  "headers": {}
                                              },
                                              "type": "endpoint"
                                          }
                                      ]
                                  }
                              ]
                          },

  {
                                  "conditions": [],
                                  "error": "DualStack is enabled but this partition does not support DualStack",
                                  "type": "error"
                              }
                          ]
                      },
                      {
                          "conditions": [],
                          "type": "tree",
                          "rules": [
                              {
                                  "conditions": [],
                                  "endpoint": {
                                      "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                      "properties": {},
                                      "headers": {}
                                  },
                                  "type": "endpoint"
                              }
                          ]
                      }
                  ]
              }
          ]
      },
      {
          "conditions": [],
          "error": "Invalid Configuration: Missing Region",
          "type": "error"
      }
  ]
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
      {
```

```
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
             "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
```

```
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {


        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
           "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint":
```

```
{
                           "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                      }
                 },
                 "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                 }
            },
            {
                 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                 "expect": {
                    "endpoint": {
                       "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                 },
                 "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                     "UseDualStack": true
                 }
            },
            {
                 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                 "expect": {
                    "endpoint": {
                       "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                 },
                 "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                 }
            },
            {
                 "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                 "expect": {
                    "endpoint":
{

                       "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
                    }
                 },
                 "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
```

```
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": true
```

```
            }
          },
          {

    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {

             "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
            "expect": {
              "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
```

```
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                }
            },
            "params": {
                "Region": "us-isob-east-1",

    "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",

        "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
```

```
                {
                    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                    "expect": {
                        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": true,
                        "UseDualStack": false,

        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }

        }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
```

```
                    "value": "Disabled",
                    "documentation": "Disabled LinuxSubscriptionsDiscovery"
                  }
                ]
              }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
              "type": "structure",
              "members": {
                "SourceRegions": {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                  "traits": {
                    "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                    "smithy.api#required": {}
                  }
                },
                "OrganizationIntegration": {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
                  "traits": {
                    "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                    "smithy.api#required": {}
                  }
                }
              },
              "traits": {
                "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
              }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
              "type":
"operation",
              "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
              },
              "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
              },
              "errors": [
                {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
                },
                {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
              }
          ],
          "traits": {
              "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
              "smithy.api#http": {

          "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Instances"
          }
      }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
          "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
              "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by
 the name of\n       <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n        <p>The valid names for this filter
are:</p>\n        <ul>\n          <li>\n             <p>\n                <code>AmiID</code>\n             </p>\n
</li>\n          <li>\n             <p>\n                <code>InstanceID</code>\n             </p>\n          </li>\n
<li>\n          <p>\n                <code>AccountID</code>\n             </p>\n          </li>\n          <li>\n
<p>\n                <code>Status</code>\n             </p>\n          </li>\n          <li>\n             <p>\n
<code>Region</code>\n             </p>\n          </li>\n          <li>\n             <p>\n
<code>UsageOperation</code>\n             </p>\n          </li>\n          <li>\n
   <p>\n                <code>ProductCode</code>\n             </p>\n          </li>\n          <li>\n             <p>\n
   <code>InstanceType</code>\n             </p>\n          </li>\n        </ul>\n        <p>The valid Operators for this
filter are:</p>\n        <ul>\n          <li>\n             <p>\n                <code>contains</code>\n             </p>\n
</li>\n          <li>\n             <p>\n                <code>equals</code>\n             </p>\n          </li>\n          <li>\n
   <p>\n                <code>Notequal</code>\n             </p>\n          </li>\n        </ul>"
              }
          },
          "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                  "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
```

```
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
            "Instances": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                "traits": {
                    "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
```

```json
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n      organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }

      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't
match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n
<li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n
<p>The valid Operators for this filter are:</p>\n
      <ul>\n        <li>\n          <p>\n            <code>contains</code>\n          </p>\n        </li>\n
<li>\n          <p>\n            <code>equals</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
      <code>Notequal</code>\n          </p>\n        </li>\n      </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
```

```
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>",
            "smithy.api#length": {


              "min": 1,
                "max": 16384
              }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "EQUAL",
            "value": "Equal",
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
```

```
              "documentation": "Not equal operator"
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
          "smithy.api#length": {
            "min": 1,
            "max": 20
          }
        }
      },

    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled OrganizationIntegration"
            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled OrganizationIntegration"
            }
          ]
        }
      },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [

{
              "name": "InProgress",
              "value": "InProgress",
              "documentation": "InProgress status"
            },
```

```
                    {
                      "name": "Completed",
                      "value": "Completed",
                      "documentation": "Completed status"
                    },
                    {
                      "name": "Successful",
                      "value": "Successful",
                      "documentation": "Successful status"
                    },
                    {
                      "name": "Failed",
                      "value": "Failed",
                      "documentation": "Failed status"
                    }
                  ]
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
            "type": "list",
            "member": {
                "target": "smithy.api#String",
                "traits":
{
                    "smithy.api#length": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                },
                "smithy.api#uniqueItems": {}
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
            "type": "map",
            "key": {
                "target": "smithy.api#String"
            },
            "value": {
                "target": "smithy.api#String"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
```

```
            "type": "structure",
            "members": {
                "Name": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The name of the subscription.</p>"

                }
                },
                "Type": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
                    }
                },
                "InstanceCount": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
                    "traits": {
                        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
                "smithy.api#error": "client"
            }
        },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },

    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required": {}
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },

          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n     subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
          "smithy.api#error": "client"
        }
      }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
```

```go
	out.ResultMetadata = metadata
	return out, nil
}

type ListLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//   - Beneficiary
	//   - ProductSKU
	//   - Fingerprint
	//
	//   - Status
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListLicensesOutput struct {

	// License details.
	Licenses []types.License

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
```

```go
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if
err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
```

```go
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:
			ServiceID,
		OperationName: "ListLicenses",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/grafana/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
```

```go
	if params == nil {
		params = &DisassociateLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*DisassociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type DisassociateLicenseInput struct {

	// The type of license to remove from the workspace.
	//
	//
	This member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to remove the Grafana Enterprise license from.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

	// A structure containing information about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
```

```
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "DisassociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
```

```go
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
 {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```go
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type
 CheckoutLicenseInput struct {

 // Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
```

```
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource
    Name (ARN) of the checkout license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}
```

```go
func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
 return err
 }
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
 return err
 }
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
 }

if err = addlegacyEndpointContextSetter(stack, options); err != nil
{
 return err
 }
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
```

```go
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack);
	err != nil {
		return err
	}
	if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
	middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
```

```go
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
	if params == nil {
		params = &CheckoutBorrowLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
	c.addOperationCheckoutBorrowLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*CheckoutBorrowLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type CheckoutBorrowLicenseInput struct {

	// Unique, case-sensitive identifier
	that you provide to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// Digital signature method. The possible value is JSON Web Signature (JWS)
	// algorithm PS384. For more information, see RFC 7518 Digital Signature with
	// RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
	//
	// This member is required.
	DigitalSignatureMethod types.DigitalSignatureMethod

	// License entitlements. Partial checkouts are not supported.
	//
	// This member is required.
	Entitlements []types.EntitlementData
```

```go
	// Amazon Resource Name (ARN) of the license. The license must use the borrow
	// consumption configuration.
	//
	// This member is required.
	LicenseArn *string

	// Information about constraints.
	CheckoutMetadata []types.Metadata

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

	// Information about constraints.
	CheckoutMetadata []types.Metadata

	// Allowed license entitlements.
	EntitlementsAllowed
	[]types.EntitlementData

	// Date and time at which the license checkout expires.
	Expiration *string

	// Date and time at which the license checkout is issued.
	IssuedAt *string

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License consumption token.
	LicenseConsumptionToken *string

	// Node ID.
	NodeId *string

	// Signed token.
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}
```

```go
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return
```

```go
  err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager
```

```go
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide
 to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity
*types.DatetimeRange

// Information about the license.
LicenseMetadata []types.Metadata

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string
```

```go
	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
		return fmt.Errorf("add
protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
```

```go
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
    err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
```

```
  Region:       region,
  ServiceID:     ServiceID,
  OperationName:
"CreateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version
```

```go
	*string

	noSmithyDocumentSerde
}

type GetLicenseOutput struct {

	// License details.
	License *types.License

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return
	err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
```

```
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
if err = addOpGetLicenseValidationMiddleware(stack);
err != nil {
 return err
 }
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
return nil
```

```
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
```

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
```

```go
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
   - ProductSKU
 //   - Status
 //   - Fingerprint
 //   - IssuerName
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense
```

```go
	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
	middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
```

```go
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack);
err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:       region,
ServiceID:    ServiceID,
OperationName: "ListReceivedLicenses",
```

```
    }
}
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
    excluding those notices that do not pertain to any part of
    the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
    distribution, then any Derivative Works that You distribute must
    include a readable copy of the attribution notices contained
    within such NOTICE file, excluding

those notices that do not

    pertain to any part of the Derivative Works, in at least one
    of the following places: within a NOTICE text file distributed
    as part of the Derivative Works; within the Source form or
    documentation, if provided along with the Derivative Works; or,
    within a display generated by the Derivative Works, if and
    wherever such third-party notices normally appear. The contents
    of the NOTICE file are for informational purposes only and
    do not modify the License. You may add Your own attribution
    notices within Derivative Works that You distribute, alongside
    or as an addendum to the NOTICE text from the Work, provided
    that such additional attribution notices cannot be construed
    as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
 incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
```

"<p>Details about an Active Directory identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n   <note>\n   <p>Your estimated bill for charges on the number of users and related
costs will take 48\n   hours to appear for billing periods that haven't closed (marked as Pending billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
   monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p>\n   </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/AssociateUser"
        },
        "smithy.api#idempotent": {}
      }

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
```

```
        },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n     resource.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
           },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
```

subscriptions.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/identity-provider/DeregisterIdentityProvider"
                },
                "smithy.api#idempotent": {}
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
            "type": "structure",
            "members": {
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Product": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                        "smithy.api#required": {}
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
            "type": "structure",
            "members": {
                "IdentityProviderSummary": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                    "traits": {
                        "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
                        "smithy.api#required": {}
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"

        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }

    },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",

```
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },

    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
```

```
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used
to return more specific results from a describe\n      operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n      attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
```

```
            "traits": {
                "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n     settings such as the subnets to provision VPC endpoints.</p>",
                "smithy.api#required":
{}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
```

```
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status
of an EC2 instance resource.</p>",
            "smithy.api#required": {}
          }
        },
        "Products": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
            "smithy.api#required": {}
          }
        },
        "LastStatusCheckDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
          }

      }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
              "smithy.api#required": {}
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          },
          "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
          },
          "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
```

```
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",

      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
```

            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
          }

        },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
            "sdkId": "License Manager User Subscriptions",
            "arnNamespace": "license-manager-user-subscriptions",
            "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-user-subscriptions"
           },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id",
              "content-length",
              "x-api-key",
              "authorization",
              "x-amz-date",
              "x-amz-security-token",
              "Access-Control-Allow-Headers",
              "Access-Control-Allow-Methods",
              "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",

```
                "x-amzn-requestid",
                "x-amzn-trace-id"


]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
                "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
                    "documentation": "The AWS region used to dispatch the request.",
                    "type": "String"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack
 endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                    "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }

        },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
```

```
                    "ref": "Endpoint"
                  }
              ]
          }
      ],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseFIPS"
                },

                true
              ]
            }
          ],
          "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
          "type": "error"
        },
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ],

"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
          "type": "error"
        },
        {
          "conditions": [],
          "endpoint": {
            "url": {
              "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
          },
```

```
                    "type": "endpoint"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [

    {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "type": "tree",

"rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },
                                        true
                                    ]
                                },
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
```

```
                            "ref": "UseDualStack"

              },
              true
          ]
      }
  ],
  "type": "tree",
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",

"argv": [

                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
          }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {

"fn": "getAttr",

            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
        }
      ]
    }
  ],
  "type": "tree",
  "rules": [
    {
```

```
                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ]
                        },
                        {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                            "type": "error"
                        }
                    ]
                },
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }
                    ],
                    "type": "tree",

        "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsFIPS"
```

```
                            ]
                          }
                        ]
                      }
                    ],
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}

          },
                            "type": "endpoint"
                        }
                      ]
                    },
                    {
                      "conditions": [],
                      "error": "FIPS is enabled but this partition does not support FIPS",
                      "type": "error"
                    }
                  ]
                },
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
```

```
                      "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {

                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                  }
                ]
              },
              {
                "conditions": [],
                "error": "DualStack is enabled but this partition does
not support DualStack",
                "type": "error"
              }
            ]
          },
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
          }
        ]
      }
    ]
```

```
            },
            {
              "conditions":
[],
              "error": "Invalid Configuration: Missing Region",
              "type": "error"
            }
          ]
        },
        "smithy.rules#endpointTests": {
          "testCases": [
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                }
              },

          "params": {
                "Region": "us-east-1",
```

```
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },

      "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
```

```
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                }
              },
              "params": {
                "Region": "cn-north-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                }
              },
              "params": {
                "Region": "cn-north-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                }
              },

      "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-gov-east-1",
```

```
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
                    }
                },

    "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                "expect":
{
                    "endpoint": {
```

```
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack":
true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }

},
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
       "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
```

```
                }
            },
            {


"documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://example.com"
                    }
                },
                "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                "expect": {
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "Missing region",
                "expect": {
                    "error": "Invalid Configuration: Missing Region"
                }
            }
        ],
        "version": "1.0"
```

```
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
              "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
              },
              "smithy.api#paginated":
{
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
              }
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
          {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
```

```
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type":
"structure",
        "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
```

```
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListProductSubscriptions"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {

  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n    more sets of key-value pairs that you specify.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },

    "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceUserSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
```

```
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
              }
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
```

```
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {

      "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }

    },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for a user.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        },
        "SubscriptionStartDate": {
```

```
                        "target": "smithy.api#String",
                        "traits": {
                            "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
                        }
                    },
                    "SubscriptionEndDate": {
                        "target": "smithy.api#String",
                        "traits": {
                            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
                        }
                    }
                },
                "traits": {
                    "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
                }
            },
            "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
                "type": "list",
                "member": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
                }
            },
            "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
                "type": "operation",
                "input": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
                },
                "output": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
                },
                "errors": [
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                    },
                    {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/RegisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n      subnets to provision VPC endpoints.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
               "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
               "smithy.api#required": {}
            }
         }
      }
   },
   "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
         "message": {
            "target": "smithy.api#String"
         }
      },
      "traits": {
         "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
         "smithy.api#error": "client",
         "smithy.api#httpError": 404
      }
   },
   "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
      "type": "string",

   "traits": {
         "smithy.api#length": {
            "min": 5,
            "max": 200
         },
         "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
      }
   },
   "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
      "type": "structure",
      "members": {
         "message": {
            "target": "smithy.api#String"
         }
      },
      "traits": {
         "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
         "smithy.api#error": "client"
      }
   },
   "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
```

```
"members": {
    "Subnets": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
            "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
            "smithy.api#length": {
                "min": 1
            },
            "smithy.api#required": {}
        }
    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n      your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n      subnets to provision VPC endpoints, and the security group ID that
 is associated with the VPC\n      endpoints. The security group should permit inbound TCP port 1688
communication from resources\n      in the VPC.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n    <note>\n      <p>Your estimated bill for charges on the number of users and related costs
will take 48\n      hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
  For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n       monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n       </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StartProductSubscription"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
```

```
                }
              },
              "Domain": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The domain name of the user.</p>"
                }
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

          },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
```

```
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required":
{}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation":
 "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {

            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
 request.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
```

```
            "type": "operation",
            "input": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
            },
            "output": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
            },
            "errors": [
               {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
               },
               {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
               {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
               },
               {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
               }
            ],
            "traits": {
               "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity\n      provider.</p>",
               "smithy.api#http": {
                  "code": 200,
                  "method": "POST",
                  "uri": "/identity-provider/UpdateIdentityProviderSettings"
               },
               "smithy.api#idempotent": {}
            }
         },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                  "smithy.api#required": {}
               }
            },
```

```
"UpdateSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n    update any combination of settings in a single operation such as the:</p>\n
<ul>\n        <li>\n            <p>Subnets which you want to add to provision VPC endpoints.</p>\n        </li>\n
    <li>\n            <p>Subnets which you want to remove the VPC endpoints from.</p>\n        </li>\n
<li>\n            <p>Security group ID which permits traffic to the VPC endpoints.</p>\n        </li>\n    </ul>",
        "smithy.api#required": {}
    }
    }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n    require connectivity to activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID
```

```
 that allows inbound TCP port 1688 communication between resources in\n     your VPC and the VPC endpoints for
activation servers.</p>"
                   }
                 }
             },
             "traits": {
                 "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as\n     the subnets to provision VPC endpoints.</p>"
             }
         },
         "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
             "type": "structure",
             "members": {
                 "message": {
                     "target": "smithy.api#String"
                 }
             },
             "traits": {
                 "smithy.api#documentation": "<p>A parameter is not valid.</p>",
                 "smithy.api#error": "client"
             }
         }
     }
}
{
   "smithy": "2.0",
   "metadata": {
     "suppressions": [
         {
             "id": "HttpMethodSemantics",
             "namespace": "*"
         },
         {
             "id": "HttpResponseCodeSemantics",
             "namespace": "*"
         },
         {
             "id": "PaginatedTrait",
             "namespace": "*"
         },
         {
             "id": "HttpHeaderTrait",
             "namespace": "*"
         },
         {
             "id": "HttpUriConflict",
             "namespace": "*"
         },
```

```
      {
        "id": "Service",
        "namespace": "*"
      }
    ]
  },
  "shapes": {
    "com.amazonaws.licensemanager#AWSLicenseManager": {
      "type": "service",
      "version": "2018-08-01",
      "operations": [
        {
          "target": "com.amazonaws.licensemanager#AcceptGrant"
        },


{
          "target": "com.amazonaws.licensemanager#CheckInLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CheckoutLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateGrant"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateGrantVersion"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
         },
        {
          "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateToken"
        },
```

```
        {
            "target": "com.amazonaws.licensemanager#DeleteGrant"
        },
        {
            "target": "com.amazonaws.licensemanager#DeleteLicense"
        },
        {
            "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
        },
        {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
        },
        {
            "target": "com.amazonaws.licensemanager#DeleteToken"
        },
        {
            "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
        }

    },
        {
            "target": "com.amazonaws.licensemanager#GetAccessToken"
        },
        {
            "target": "com.amazonaws.licensemanager#GetGrant"
        },
        {
            "target": "com.amazonaws.licensemanager#GetLicense"
        },
        {
            "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
        },
        {
            "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
        },
        {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
        },
        {
            "target": "com.amazonaws.licensemanager#GetLicenseUsage"
        },
        {
            "target": "com.amazonaws.licensemanager#GetServiceSettings"
        },
        {
            "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
         },
        {
            "target": "com.amazonaws.licensemanager#ListDistributedGrants"
```

```
    },
    {
      "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
    },
    {

      "target": "com.amazonaws.licensemanager#ListLicenseVersions"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListResourceInventory"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTokens"
    },
    {
      "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
    },
    {
```

```
          "target": "com.amazonaws.licensemanager#RejectGrant"
        },
        {
          "target": "com.amazonaws.licensemanager#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanager#UntagResource"
        },
        {
          "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
        },
        {
          "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
        },
        {
          "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
        },
        {
          "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager",
          "arnNamespace": "license-manager",
          "cloudFormationName": "LicenseManager",
          "cloudTrailEventSource": "licensemanager.amazonaws.com",
          "endpointPrefix": "license-manager"
        },
        "aws.auth#sigv4": {
          "name": "license-manager"
        },
        "aws.protocols#awsJson1_1": {},
        "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n        Amazon Web Services accounts and on-premises servers.</p>",
        "smithy.api#title": "AWS License Manager",
        "smithy.api#xmlNamespace": {
          "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
        },
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
```

```
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "Boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
                "type": "Boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "String"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },
                                    true
```

```
                    ]
                }
            ],
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
            "type": "error"
        },
        {

"conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
            "type": "error"
        },
        {
            "conditions": [],
            "endpoint": {
                "url": {
                    "ref": "Endpoint"
                },
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ]
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ],
    "type": "tree",
    "rules": [
```

```
{
    "conditions": [
        {
            "fn": "aws.partition",
            "argv": [
                {
                    "ref": "Region"
                }
            ],
            "assign": "PartitionResult"
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref":
"UseFIPS"
                        },
                        true
                    ]
                },
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
```

                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                      ]
                  }
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  true,
                  {
                    "fn": "getAttr",
                    "argv": [
                      {
                        "ref": "PartitionResult"
                      },

    "supportsDualStack"
                    ]
                  }
                ]
              }
            ],
            "type": "tree",
            "rules": [
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                  "properties": {},
                  "headers": {}

                },
                "type": "endpoint"
              }
            ]
          },
          {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
            "type": "error"
          }
        ]
      },

```
{
    "conditions": [
        {
            "fn": "booleanEquals",

            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,

                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        },
```

```
                                  {
                                     "conditions": [],
                                     "error": "FIPS is enabled but this
partition does not support FIPS",
                                     "type": "error"
                                  }
                               ]
                            },
                            {
                               "conditions": [
                                  {
                                     "fn": "booleanEquals",
                                     "argv": [
                                        {
                                           "ref": "UseDualStack"
                                        },
                                        true
                                     ]
                                  }
                               ],
                               "type": "tree",
                               "rules": [
                                  {
                                     "conditions": [
                                        {
                                           "fn": "booleanEquals",
                                           "argv": [
                                              true,
                                              {
                                                 "fn": "getAttr",
                                                 "argv": [
                                                    {
                                                       "ref": "PartitionResult"
                                                    },

"supportsDualStack"

                                                 ]
                                              }
                                           ]
                                        }
                                     ],
                                     "type": "tree",
                                     "rules": [
                                        {
                                           "conditions": [],
                                           "endpoint": {
                                              "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
```

```
                    "properties": { },
                        "headers": { }

                },
                    "type": "endpoint"
                }
              ]
            },
            {
              "conditions": [],
              "error": "DualStack is enabled but this partition does not support DualStack",
              "type": "error"
            }
          ]
        },
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
             "properties": { },
            "headers": { }
          },
          "type": "endpoint"
        }
      ]
    }
  ]
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint":
{
             "url": "https://license-manager.af-south-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "af-south-1",
          "UseFIPS": false,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-3.amazonaws.com"
            }
```

```
        },
        "params": {
          "Region": "ap-northeast-3",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-south-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-south-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-southeast-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-southeast-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-southeast-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-southeast-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
```

```
                    {

      "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager.ap-southeast-3.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "ap-southeast-3",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager.ca-central-1.amazonaws.com"
                  }
                },

               "params": {
                  "Region": "ca-central-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager.eu-central-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "eu-central-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager.eu-north-1.amazonaws.com"
                  }
```

```
          },
          "params": {
            "Region": "eu-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region":
"eu-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-west-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
          "expect": {

            "endpoint": {
              "url": "https://license-manager.eu-west-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-west-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
```

```
{
    "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager.eu-west-3.amazonaws.com"
      }
    },
    "params": {
      "Region": "eu-west-3",
      "UseFIPS": false,

"UseDualStack": false
    }
  },
  {
    "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager.me-south-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "me-south-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager.sa-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "sa-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager.us-east-1.amazonaws.com"
      }
    },
    "params": {
```

```
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }

      },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager.us-east-2.amazonaws.com"
                  }

              },
                "params": {
                  "Region": "us-east-2",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-fips.us-east-2.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-2",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region
```

us-west-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager.us-west-1.amazonaws.com"
                        }
                    },
                    "params": {
                        "Region": "us-west-1",
                        "UseFIPS": false,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-fips.us-west-1.amazonaws.com"
                        }
                    },
                    "params": {

            "Region": "us-west-1",
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager.us-west-2.amazonaws.com"
                        }
                    },
                    "params": {
                        "Region": "us-west-2",
                        "UseFIPS": false,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-fips.us-west-2.amazonaws.com"
                        }
                    },
                    "params": {

```
            "Region": "us-west-2",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,


            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
```

```
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
                        }
                    },
                    "params": {
                        "Region": "cn-northwest-1",
                        "UseFIPS": false,
                        "UseDualStack": false
                    }
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
```

```
              "UseDualStack": true
            }
        },

  {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-gov-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-gov-west-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-gov-west-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
        },
        {
            "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
```

```
                                "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
                            }
                        },
                        "params": {
                            "Region": "us-gov-west-1",
                            "UseFIPS": true,
                            "UseDualStack": false
                        }
                    },
                    {
                        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-fips.us-gov-east-1.api.aws"
                            }
                        },

        "params": {
                            "Region": "us-gov-east-1",
                            "UseFIPS": true,
                            "UseDualStack": true
                        }
                    },
                    {
                        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager.us-gov-east-1.api.aws"
                            }
                        },
                        "params": {
                            "Region": "us-gov-east-1",
                            "UseFIPS": false,
                            "UseDualStack": true
                        }
                    },
                    {
                        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
                            }
                        },
                        "params": {
                            "Region": "us-iso-east-1",
                            "UseFIPS": false,
                            "UseDualStack": false
```

```
                  }
               },
               {
                  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                  "expect": {
                     "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                  },
                  "params": {
                     "Region": "us-iso-east-1",
                     "UseFIPS": true,
                      "UseDualStack": true
                  }
               },
               {
                  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                     "endpoint": {
                        "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
                     }
                  },
                  "params": {
                     "Region": "us-iso-east-1",
                     "UseFIPS": true,
                     "UseDualStack": false
                  }
               },
               {
                  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                     "error": "DualStack is enabled
but this partition does not support DualStack"
                  },
                  "params": {
                     "Region": "us-iso-east-1",
                     "UseFIPS": false,
                     "UseDualStack": true
                  }
               },
               {
                  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                     "endpoint": {
                        "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
                     }
                  },
                  "params": {
                     "Region": "us-isob-east-1",
                     "UseFIPS": false,
```

```
          "UseDualStack": false
        }
      },

    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
          }
        },

    "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

    "expect": {
          "endpoint": {
            "url": "https://example.com"
          }
        },
```

```
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,

    "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error":
  "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
```

```
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanager#AcceptGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#AcceptGrantRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
    }
},
"com.amazonaws.licensemanager#AcceptGrantRequest": {
```

```
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant
ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#AccessDeniedException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
```

            "traits": {
                "aws.protocols#awsQueryError": {

"code": "ServiceAccessDenied",
                    "httpResponseCode": 401
                },
                "smithy.api#documentation": "<p>Access to resource denied.</p>",
                "smithy.api#error": "client",
                "smithy.api#httpError": 401
            }
        },
        "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
            "type": "enum",
            "members": {
                "DISTRIBUTED_GRANTS_ONLY": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
                    }
                },
                "ALL_GRANTS_PERMITTED_BY_ISSUER": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#AllowedOperation": {

"type": "enum",
            "members": {
                "CREATE_GRANT": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "CreateGrant"
                    }
                },
                "CHECKOUT_LICENSE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "CheckoutLicense"
                    }
                },
                "CHECKOUT_BORROW_LICENSE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "CheckoutBorrowLicense"

```
        }
      },
      "CHECK_IN_LICENSE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "CheckInLicense"
        }
      },
      "EXTEND_CONSUMPTION_LICENSE":
{

        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "ExtendConsumptionLicense"
        }
      },
      "LIST_PURCHASED_LICENSES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "ListPurchasedLicenses"
        }
      },
      "CREATE_TOKEN": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "CreateToken"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#AllowedOperationList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#AllowedOperation"
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max":
7
      }
    }
  },
  "com.amazonaws.licensemanager#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 2048
```

```
        },
        "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z-
z0-9_/.-]{0,62}:[A-Za-
z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
      }
    },
    "com.amazonaws.licensemanager#ArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
      }
    },
    "com.amazonaws.licensemanager#AuthorizationException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError":
  {
          "code": "AuthorizationFailure",
          "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n          policy associated with this account.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 403
      }
    },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
      "type": "structure",
      "members": {
        "LastRunTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes automated discovery.</p>"
      }
    },

  "com.amazonaws.licensemanager#Boolean": {
      "type": "boolean",
      "traits": {
```

```machine_data
          "smithy.api#default": false
        }
      },
      "com.amazonaws.licensemanager#BorrowConfiguration": {
        "type": "structure",
        "members": {
          "AllowEarlyCheckIn": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxTimeToLiveInMinutes": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
              "smithy.api#required": {}
            }
          }

      },
        "traits": {
          "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#BoxBoolean": {
        "type": "boolean"
      },
      "com.amazonaws.licensemanager#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanager#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanager#CheckInLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
```

```
          "target": "com.amazonaws.licensemanager#AuthorizationException"

      },
        {
          "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
      }
    },
    "com.amazonaws.licensemanager#CheckInLicenseRequest":
{
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
```

```
    },
    "com.amazonaws.licensemanager#CheckInLicenseResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
          "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
```

          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
        "type": "structure",
        "members":
 {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
              "smithy.api#required": {}
            }
          },
          "DigitalSignatureMethod": {
            "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
            "traits": {
              "smithy.api#documentation": "<p>Digital signature method. The possible
 value is JSON Web Signature (JWS) algorithm PS384.\n   For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS.</p>",
              "smithy.api#required": {}
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          },
          "CheckoutMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about constraints.</p>"
            }

```
                },
                "ClientToken": {
                    "target": "com.amazonaws.licensemanager#ClientToken",
                    "traits": {
                        "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
                    }
                },
                "LicenseConsumptionToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License consumption token.</p>"
                    }
                },
                "EntitlementsAllowed":
{
                    "target": "com.amazonaws.licensemanager#EntitlementDataList",
                    "traits": {
                        "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
                    }
                },
                "NodeId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Node ID.</p>"
                    }
                },
                "SignedToken": {
                    "target": "com.amazonaws.licensemanager#SignedToken",
                    "traits": {
                        "smithy.api#documentation": "<p>Signed token.</p>"
                    }
                },
```

```
      "IssuedAt": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
        }
      },
      "Expiration": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
        }
      },
      "CheckoutMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
          "smithy.api#documentation": "<p>Information about constraints.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#CheckoutLicense": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
```

```
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n          <p>If the
account that created the license is the same that is performing the check out, you must\n          specify the account
as the beneficiary.</p>\n        </note>"
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
      "type": "structure",
      "members": {
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product
SKU.</p>",
            "smithy.api#required": {}
          }
        },
        "CheckoutType": {
          "target": "com.amazonaws.licensemanager#CheckoutType",
          "traits": {
            "smithy.api#documentation": "<p>Checkout type.</p>",
            "smithy.api#required": {}
          }
        },
        "KeyFingerprint": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
            "smithy.api#required": {}
          }
        },
```

```
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        }
      },
      "traits": {

    "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
      "type": "structure",
      "members": {
        "CheckoutType": {
          "target": "com.amazonaws.licensemanager#CheckoutType",
          "traits": {
            "smithy.api#documentation": "<p>Checkout type.</p>"
          }
        },
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
```

```
          },
          "EntitlementsAllowed": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
            }
          },
          "SignedToken": {
            "target": "com.amazonaws.licensemanager#SignedToken",
            "traits": {
              "smithy.api#documentation": "<p>Signed token.</p>"
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          },
          "IssuedAt": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
            }
          },
          "Expiration": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutType": {
        "type": "enum",
        "members": {
          "PROVISIONAL": {
            "target": "smithy.api#Unit",
```

```
            "traits": {
                "smithy.api#enumValue": "PROVISIONAL"
            }
        },
        "PERPETUAL": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "PERPETUAL"
            }
        }
    }
},
"com.amazonaws.licensemanager#ClientRequestToken": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 36
        }
    }
},
"com.amazonaws.licensemanager#ClientToken": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 2048
        },
        "smithy.api#pattern": "^\\S+$"
    }
},
"com.amazonaws.licensemanager#ConflictException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
    },
    "traits": {
        "aws.protocols#awsQueryError": {
            "code": "ConflictException",
            "httpResponseCode":
409
        },
        "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 409
```

```
          }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
          "type": "structure",
          "members": {
            "ResourceType": {
              "target": "com.amazonaws.licensemanager#ResourceType",
              "traits": {
                "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
              }
            },
            "ConsumedLicenses": {
              "target": "com.amazonaws.licensemanager#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
              }
            }
          },

  "traits": {
          "smithy.api#documentation": "<p>Details about license consumption.</p>"
        }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
          }
        },
        "com.amazonaws.licensemanager#ConsumptionConfiguration": {
          "type": "structure",
          "members": {
            "RenewType": {
              "target": "com.amazonaws.licensemanager#RenewType",
              "traits": {
                "smithy.api#documentation": "<p>Renewal frequency.</p>"
              }
            },
            "ProvisionalConfiguration": {
              "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
              "traits": {
                "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
              }
            },
            "BorrowConfiguration": {
              "target": "com.amazonaws.licensemanager#BorrowConfiguration",
              "traits": {
                "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
```

```
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target":
"com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrantRequest": {
```

```
    "type": "structure",
    "members": {
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientToken",
        "traits": {
          "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant name.</p>",
          "smithy.api#required": {}

  }
    },
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
        "smithy.api#required": {}
      }
    },
    "Principals": {
      "target": "com.amazonaws.licensemanager#PrincipalArnList",
      "traits": {
        "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n        (ARN):</p>\n    <ul>\n        <li>\n            <p>An Amazon Web Services
account, which includes only the account specified.</p>\n        </li>\n    </ul>\n    <ul>\n        <li>\n
   <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n          </li>\n        </ul>\n
<ul>\n
       <li>\n            <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n        </ul>",
        "smithy.api#required": {}
      }
    },
    "HomeRegion": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Home Region of the grant.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowedOperations": {
      "target": "com.amazonaws.licensemanager#AllowedOperationList",
      "traits": {
```

```
            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },

    "com.amazonaws.licensemanager#CreateGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersion":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
  }
},
"com.amazonaws.licensemanager#CreateGrantVersionRequest": {
  "type": "structure",
  "members": {
    "ClientToken": {
      "target": "com.amazonaws.licensemanager#ClientToken",
      "traits": {
        "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

  "smithy.api#required": {}
      }
    },
    "GrantArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
        "smithy.api#required": {}
      }
    },
    "GrantName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
```

```
              "smithy.api#documentation": "<p>Grant name.</p>"
            }
          },
          "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>Grant status reason.</p>"
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the grant.</p>"
            }
          },
          "Options": {
            "target": "com.amazonaws.licensemanager#Options",
            "traits": {
              "smithy.api#documentation": "<p>The options specified for the grant.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    }
  },
  "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Grant ARN.</p>"
        }
      },
      "Status": {
```

```
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>New version of the grant.</p>"
                    }
                }
            },
            "traits":
{
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicense": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RedirectException"
                 },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
```

```
      "traits": {
        "smithy.api#documentation": "<p>Creates a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager. Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or\n         all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of licenses purchased and used.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n          blocks the launch of new instances.</p>"
          }
        },
        "LicenseRules": {
          "target":
 "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n          vary by dimension, as follows.</p>\n
<ul>\n          <li>\n            <p>\n              <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
     <code>licenseAffinityToHost</code> |\n              <code>maximumCores</code> |
<code>minimumCores</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n              <code>maximumCores</code> |
<code>minimumCores</code> |\n              <code>maximumSockets</code> | <code>minimumSockets</code> |\n
     <code>maximumVcpus</code> | <code>minimumVcpus</code>\n            </p>\n          </li>\n
<li>\n
  <p>\n              <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n              <code>maximumSockets</code> |
<code>minimumSockets</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n              <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n            </p>\n          </li>\n          </ul>\n          <p>The unit for
```

<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for <code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n <code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n <code>True</code> and <code>False</code>.</p>"

```
                }
            },

    "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                    "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
                }
            },
            "DisassociateWhenNotFound": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            },
            "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                    "smithy.api#documentation": "<p>Product information.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
        "type": "operation",
```

```
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {

            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n      For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
```

the <i>License Manager User Guide</i>.</p>",
                    "smithy.api#required": {}
                }
            },
            "DestinationLicenseContext":
{
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI in
the <i>License Manager User Guide</i>.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

                "traits": {
                    "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {

```
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                 },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Creates a report generator.</p>"
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
            "type": "structure",
            "members": {
                "ReportGeneratorName":
{
                    "target": "com.amazonaws.licensemanager#ReportGeneratorName",
                    "traits": {
                        "smithy.api#documentation": "<p>Name of the report generator.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Type": {
                    "target": "com.amazonaws.licensemanager#ReportTypeList",
                    "traits": {
                        "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n        <ul>\n            <li>\n                <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n            </li>\n            <li>\n                <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n            </li>\n
</ul>",
                        "smithy.api#required": {}

                    }
                },
```

```
      "ReportContext": {
        "target": "com.amazonaws.licensemanager#ReportContext",
        "traits": {
          "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
          "smithy.api#required": {}
        }
      },
      "ReportFrequency": {
        "target": "com.amazonaws.licensemanager#ReportFrequency",
        "traits": {
          "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
          "smithy.api#required": {}
        }
      },
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientRequestToken",
        "traits": {
          "smithy.api#documentation": "<p>Unique,
 case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
          "smithy.api#required": {}
        }
      },
      "Description": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Description of the report generator.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanager#TagList",
        "traits": {
          "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
    "type": "structure",
    "members": {
      "LicenseManagerReportGeneratorArn":
{
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
```

```
                      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
                    }
                  }
                },
                "traits": {
                  "smithy.api#output": {}
                }
              },
              "com.amazonaws.licensemanager#CreateLicenseRequest": {
                "type": "structure",
                "members": {
                  "LicenseName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                      "smithy.api#documentation": "<p>License name.</p>",
                      "smithy.api#required": {}
                    }
                  },
                  "ProductName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                      "smithy.api#documentation": "<p>Product
name.</p>",
                      "smithy.api#required": {}
                    }
                  },
                  "ProductSKU": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                      "smithy.api#documentation": "<p>Product SKU.</p>",
                      "smithy.api#required": {}
                    }
                  },
                  "Issuer": {
                    "target": "com.amazonaws.licensemanager#Issuer",
                    "traits": {
                      "smithy.api#documentation": "<p>License issuer.</p>",
                      "smithy.api#required": {}
                    }
                  },
                  "HomeRegion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                      "smithy.api#documentation": "<p>Home Region for the license.</p>",
                      "smithy.api#required": {}
```

```
          }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>",
            "smithy.api#required": {}
          }
        },
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n          running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n          usage.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        }
      },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersion": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#ConflictException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target":
"com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>",
            "smithy.api#required": {}
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#Issuer",
```

```
                "traits":
{
                    "smithy.api#documentation": "<p>License issuer.</p>",
                    "smithy.api#required": {}
                }
            },
            "HomeRegion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Home Region of the license.</p>",
                    "smithy.api#required": {}
                }
            },
            "Validity": {
                "target": "com.amazonaws.licensemanager#DatetimeRange",
                "traits": {
                    "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license.</p>"
                }
            },
            "Entitlements": {
                "target": "com.amazonaws.licensemanager#EntitlementList",
                "traits": {
                    "smithy.api#documentation": "<p>License entitlements.</p>",
                    "smithy.api#required": {}
                }
            },
            "ConsumptionConfiguration": {
                "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
                    "smithy.api#required": {}
                }
            },

    "Status": {
                "target": "com.amazonaws.licensemanager#LicenseStatus",
                "traits": {
                    "smithy.api#documentation": "<p>License status.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the license.</p>"
            }
          }
        },
        "traits": {

"smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New version of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
```

```
{}
        }
    },
    "com.amazonaws.licensemanager#CreateToken": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateTokenRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateTokenResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },

            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateTokenRequest": {
        "type": "structure",
        "members": {
            "LicenseArn": {
```

```
              "target": "com.amazonaws.licensemanager#Arn",
              "traits":
 {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
                  "smithy.api#required": {}
              }
          },
          "RoleArns": {
              "target": "com.amazonaws.licensemanager#ArnList",
              "traits": {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n        License Manager does not check whether the roles are in use.</p>"
              }
          },
          "ExpirationInDays": {
              "target": "com.amazonaws.licensemanager#Integer",
              "traits": {
                  "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
              }
          },
          "TokenProperties": {
               "target": "com.amazonaws.licensemanager#MaxSize3StringList",
              "traits": {
                  "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n        to the amr claim of the JWT token.</p>"
              }
          },
          "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientToken",
              "traits": {
                  "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
                  "smithy.api#required": {}
              }
          }
      },
      "traits": {
          "smithy.api#input": {}
      }
  },
  "com.amazonaws.licensemanager#CreateTokenResponse": {
      "type": "structure",
      "members": {
          "TokenId": {
              "target": "com.amazonaws.licensemanager#String",
```

```
        "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
        }
    },
    "TokenType": {
        "target": "com.amazonaws.licensemanager#TokenType",
        "traits": {
            "smithy.api#documentation": "<p>Token type.</p>"
        }
    },
    "Token": {
        "target": "com.amazonaws.licensemanager#TokenString",
        "traits": {
            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licensemanager#DateTime": {
    "type": "timestamp"
},
"com.amazonaws.licensemanager#DatetimeRange": {
    "type": "structure",
    "members": {
        "Begin": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>Start of the time range.</p>",
                "smithy.api#required": {}
            }
        },
        "End": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>End of the time range.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
    }
},
"com.amazonaws.licensemanager#DeleteGrant": {
    "type": "operation",
    "input": {
```

```
          "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          },
          "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
            }
```

```
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the
grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation":
"<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
```

```
        },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        }

},
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
      },

    "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
```

```
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
    }
  },
  "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
    "type": "structure",
    "members":
{
      "LicenseConfigurationArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the license configuration.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
    },
    "errors": [
      {
```

```
                  "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
                  "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
              {
                  "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                  "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                  "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
              },
              {
                  "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
              },
              {
                  "target": "com.amazonaws.licensemanager#ServerInternalException"
              },
              {
                  "target": "com.amazonaws.licensemanager#ValidationException"
              }
          ],
          "traits": {
              "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
          }
      },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
          "type": "structure",
          "members": {
              "LicenseManagerReportGeneratorArn": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                      "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
                      "smithy.api#required": {}
                  }
              }
          },
          "traits": {

  "smithy.api#input": {}
          }
      },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
```

```
                    "type": "structure",
                    "members": {},
                    "traits": {
                        "smithy.api#output": {}
                    }
                }
            },
            "com.amazonaws.licensemanager#DeleteLicenseRequest": {
                "type": "structure",
                "members": {
                    "LicenseArn": {
                        "target": "com.amazonaws.licensemanager#Arn",
                        "traits": {
                            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                            "smithy.api#required": {}
                        }
                    },
                    "SourceVersion": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Current version of the license.</p>",
                            "smithy.api#required":
{}
                        }
                    }
                },
                "traits": {
                    "smithy.api#input": {}
                }
            },
            "com.amazonaws.licensemanager#DeleteLicenseResponse": {
                "type": "structure",
                "members": {
                    "Status": {
                        "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
                        "traits": {
                            "smithy.api#documentation": "<p>License status.</p>"
                        }
                    },
                    "DeletionDate": {
                        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                        "traits": {
                            "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
                        }
                    }
                },
                "traits": {
                    "smithy.api#output": {}
                }
```

```
    },
    "com.amazonaws.licensemanager#DeleteToken": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },

        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenRequest": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>",
            "smithy.api#required": {}
          }
        }
```

```
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}


      }
    },
    "com.amazonaws.licensemanager#DigitalSignatureMethod": {
      "type": "enum",
      "members": {
        "JWT_PS384": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "JWT_PS384"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#Entitlement": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement name.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#Long",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
          }
        },
        "Overage": {
```

```
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
        }
      },
      "Unit": {
        "target": "com.amazonaws.licensemanager#EntitlementUnit",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement unit.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowCheckIn": {
        "target":
"com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
    }
  },
  "com.amazonaws.licensemanager#EntitlementData": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement data name.</p>",
          "smithy.api#required": {}
        }
      },
      "Value": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement
data value.</p>"
        }
      },
      "Unit": {
        "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement data unit.</p>",
          "smithy.api#required": {}
        }
      }
```

```
      },
      "traits": {
        "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementDataList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementData"
      }
    },
    "com.amazonaws.licensemanager#EntitlementDataUnit": {
      "type": "enum",
      "members": {
        "COUNT": {
          "target": "smithy.api#Unit",
          "traits": {

"smithy.api#enumValue": "Count"
          }
        },
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "SECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Seconds"
          }
        },
        "MICROSECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Microseconds"
          }
        },
        "MILLISECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Milliseconds"
          }
        },
        "BYTES": {
          "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS":
{
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits"
      }
    },
    "KILOBITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobits"
      }
    },
    "MEGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits"
      }
    },
    "GIGABITS": {
```

```
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabits"
      }
    },
    "TERABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabits"

}
    },
    "PERCENT": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Percent"
      }
    },
    "BYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes/Second"
      }
    },
    "KILOBYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes/Second"
      }
    },
    "MEGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes/Second"
      }
    },
    "GIGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
       "traits": {
        "smithy.api#enumValue": "Gigabytes/Second"
      }
    },
    "TERABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes/Second"
      }
    },
```

```json
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
          }
        },
        "TERABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits/Second"
          }
        },
        "COUNT_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count/Second"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Entitlement"
      }
    },
    "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
      "type": "structure",
```

```
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#EntitlementUnit": {
        "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "SECONDS":
{
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Seconds"
            }
          },
          "MICROSECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Microseconds"
            }
          },
          "MILLISECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Milliseconds"
            }
          },
          "BYTES": {
            "target": "smithy.api#Unit",
            "traits": {
```

```
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits"
      }
    },
    "KILOBITS": {
      "target": "smithy.api#Unit",
      "traits": {


       "smithy.api#enumValue": "Kilobits"
      }
    },
    "MEGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits"
      }
    },
    "GIGABITS": {
      "target": "smithy.api#Unit",
```

```
        "traits": {
          "smithy.api#enumValue": "Gigabits"
        }
      },
      "TERABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits"
        }
      },
      "PERCENT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Percent"
        }
      },
      "BYTES_PER_SECOND": {
        "target":
"smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes/Second"
        }
      },
      "KILOBYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes/Second"
        }
      },
      "MEGABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes/Second"
        }
      },
      "GIGABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes/Second"
        }
      },
      "TERABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {

"smithy.api#enumValue": "Terabytes/Second"
        }
      },
```

```
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
          }
        },

    "TERABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits/Second"
          }
        },
        "COUNT_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count/Second"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsage": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage name.</p>",
            "smithy.api#required": {}
          }
```

```
        },
        "ConsumedValue": {
          "target": "com.amazonaws.licensemanager#String",

        "traits": {
            "smithy.api#documentation": "<p>Resource usage consumed.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList":
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementUsage"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
         },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>",
            "smithy.api#required": {}

   }
        },
        "DryRun": {
          "target": "com.amazonaws.licensemanager#Boolean",
          "traits": {
            "smithy.api#default": false,
            "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
      "type": "structure",
      "members": {
```

```
      "LicenseConsumptionToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License consumption token.</p>"
        }
      },

      "Expiration": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#FailedDependencyException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      },
      "ErrorCode": {
        "target": "com.amazonaws.licensemanager#String"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "FailedDependency",
        "httpResponseCode": 424
      },
      "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 424
    }
  },
  "com.amazonaws.licensemanager#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#FilterName",
        "traits": {
          "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
        }
      },
```

```
      "Values": {
        "target": "com.amazonaws.licensemanager#FilterValues",
        "traits": {
          "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n       describe operation. Filters can be used
to match a set of resources by specific criteria,\n       such as tags, attributes, or IDs.</p>"
    }
  },
  "com.amazonaws.licensemanager#FilterLimitExceededException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "FilterLimitExceeded",
        "httpResponseCode": 400
      },
      "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Filter"
    }
  },
  "com.amazonaws.licensemanager#FilterName": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValue": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValues": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#FilterValue",
      "traits": {
```

```
                    "smithy.api#xmlName": "item"
                }
            }
        },
        "com.amazonaws.licensemanager#Filters": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#Filter",
                "traits": {
                    "smithy.api#xmlName": "item"
                }
            }
        },
        "com.amazonaws.licensemanager#GetAccessToken": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
 tokens\n        are valid for one hour.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetAccessTokenRequest": {
            "type": "structure",
            "members": {
                "Token": {
```

```
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
            "smithy.api#required": {}
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
            "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
      "type": "structure",
      "members": {
        "AccessToken": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Temporary access token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetGrantResponse": {
        "type": "structure",
        "members": {
          "Grant": {
            "target": "com.amazonaws.licensemanager#Grant",
            "traits": {
```

```
                "smithy.api#documentation": "<p>Grant details.</p>"
            }
        }
    },
    "traits": {

"smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#GetLicense": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"

    },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
    }
},
"com.amazonaws.licensemanager#GetLicenseConfiguration": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
    },
    "output": {
```

```
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits":
{
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurationId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
```

```
        }
      },
      "LicenseConfigurationArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
        }
      },
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Name of the license configuration.</p>"
        }
      },
      "Description": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Description of the license configuration.</p>"
        }
      },
      "LicenseCountingType": {
        "target": "com.amazonaws.licensemanager#LicenseCountingType",
        "traits": {
          "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",

        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
```

```
            "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License configuration status.</p>"
          }
        }

      },
        "OwnerAccountId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
          }
        },
        "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
          }
        },
        "ManagedResourceSummaryList": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>Product information.</p>"
          }
        },
        "AutomatedDiscoveryInformation": {
          "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
          "traits": {
            "smithy.api#documentation": "<p>Automated discovery information.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
```

```json
          "traits": {
                "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTask": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {

"target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
```

```
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                "traits": {
                    "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
                }
            },
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
                }
            },
            "SourceLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
                }
            },
            "DestinationLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
                }
```

```
                },

    "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                }
            },
            "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

            },
            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
            "LicenseManagerReportGeneratorArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
                    "smithy.api#required":
{}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {
            "ReportGenerator": {
                "target": "com.amazonaws.licensemanager#ReportGenerator",
                "traits": {
                    "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
```

```
    },
    "com.amazonaws.licensemanager#GetLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseResponse": {
      "type": "structure",
      "members": {
        "License": {
          "target": "com.amazonaws.licensemanager#License",
          "traits": {
            "smithy.api#documentation": "<p>License details.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },

    "com.amazonaws.licensemanager#GetLicenseUsage": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
      },
      "errors": [
        {
```

```
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {

    "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
        "type": "structure",
        "members": {
          "LicenseUsage": {
            "target":
"com.amazonaws.licensemanager#LicenseUsage",
            "traits": {
              "smithy.api#documentation": "<p>License usage details.</p>"
            }
          }
```

```
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "S3BucketArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"
```

```
          }
        },
        "SnsTopicArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
          }
        },
        "OrganizationConfiguration": {
          "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n         cross-account discovery.</p>"
          }
        },
        "EnableCrossAccountsDiscovery": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
          }

      },
        "LicenseManagerResourceShareArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n         provides member accounts with access to this share.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Grant": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
```

```
        "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant name.</p>",
                    "smithy.api#required": {}
                }
            },
            "ParentArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Parent ARN.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>License ARN.</p>",
                    "smithy.api#required": {}
                }
            },
            "GranteePrincipalArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {

    "smithy.api#documentation": "<p>The grantee principal ARN.</p>",
                    "smithy.api#required": {}
                }
            },
            "HomeRegion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantStatus": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>",
                    "smithy.api#required": {}
                }
            },
            "StatusReason": {
                "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status reason.</p>"
                }
            },
```

```
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant version.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
        "traits": {
          "smithy.api#documentation": "<p>Granted operations.</p>",
          "smithy.api#required": {}
        }
      },
      "Options": {
        "target": "com.amazonaws.licensemanager#Options",
        "traits": {
          "smithy.api#documentation": "<p>The options specified for the grant.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>Describes a grant.</p>"
    }
  },
  "com.amazonaws.licensemanager#GrantList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Grant"
    }
  },
  "com.amazonaws.licensemanager#GrantStatus": {
    "type": "enum",
    "members": {
      "PENDING_WORKFLOW": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_WORKFLOW"
        }
      },
      "PENDING_ACCEPT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_ACCEPT"
        }
      },
      "REJECTED": {
```

```
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"

        }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits":
{

            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
      "type": "structure",
      "members": {
```

```
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>"
          }

      },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#IssuerDetails",
          "traits": {
            "smithy.api#documentation": "<p>Granted license issuer.</p>"
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

        }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>Granted license status.</p>"
          }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
```

ISO8601-UTC format.</p>"
                }
            },
            "Beneficiary": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
                }
            },
            "Entitlements": {
                "target": "com.amazonaws.licensemanager#EntitlementList",
                "traits": {
                    "smithy.api#documentation": "<p>License entitlements.</p>"
                }
            },
            "ConsumptionConfiguration": {
                "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
                }
            },
            "LicenseMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Granted license metadata.</p>"
                }
            },
            "CreateTime": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

}
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Version of the granted license.</p>"
                }
            },
            "ReceivedMetadata": {
                "target": "com.amazonaws.licensemanager#ReceivedMetadata",
                "traits": {
                    "smithy.api#documentation": "<p>Granted license received metadata.</p>"
                }
            }
        },
        "traits": {

```
      "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
    }
  },
  "com.amazonaws.licensemanager#GrantedLicenseList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#GrantedLicense"
    }
  },
  "com.amazonaws.licensemanager#ISO8601DateTime":
{
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
    }
  },
  "com.amazonaws.licensemanager#Integer": {
    "type": "integer"
  },
  "com.amazonaws.licensemanager#InvalidParameterValueException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "InvalidParameterValueProvided",
        "httpResponseCode": 400
      },
      "smithy.api#documentation":
"<p>One or more parameter values are not valid.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#InvalidResourceStateException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
```

```
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InvalidResourceState",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },

    "com.amazonaws.licensemanager#InventoryFilter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Condition": {
          "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
          "traits": {
            "smithy.api#documentation": "<p>Condition of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }

      },
      "traits": {
        "smithy.api#documentation": "<p>An inventory filter.</p>"
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterCondition": {
      "type": "enum",
      "members": {
        "EQUALS": {
          "target": "smithy.api#Unit",
```

```
          "traits": {
            "smithy.api#enumValue": "EQUALS"
          }
        },
        "NOT_EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "NOT_EQUALS"
          }
        },
        "BEGINS_WITH": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "BEGINS_WITH"
          }
        },
        "CONTAINS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CONTAINS"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#InventoryFilter"
      }
    },
    "com.amazonaws.licensemanager#Issuer": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Issuer name.</p>",
            "smithy.api#required": {}
          }
        },
        "SignKey": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
          }
        }
```

```
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#IssuerDetails": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Issuer name.</p>"
          }
        },
        "SignKey": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
          }
        },
        "KeyFingerprint": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#License": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {

      "smithy.api#documentation": "<p>License name.</p>"
          }
```

```
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product name.</p>"
            }
          },
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product SKU.</p>"
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#IssuerDetails",
            "traits": {
              "smithy.api#documentation": "<p>License issuer.</p>"
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

"smithy.api#documentation": "<p>Home Region of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          },
          "Entitlements": {
```

```
        "target": "com.amazonaws.licensemanager#EntitlementList",
                "traits": {
                    "smithy.api#documentation": "<p>License entitlements.</p>"
                }
            },
            "ConsumptionConfiguration": {
                "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
                }
            },
            "LicenseMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>License metadata.</p>"
                }
            },
            "CreateTime": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>License
creation time.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License version.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseConfiguration": {
        "type": "structure",
        "members": {
            "LicenseConfigurationId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
                }
            },
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
```

```
        }
      },
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Name of the license configuration.</p>"
        }
      },
      "Description": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Description of the license configuration.</p>"
        }
      },
      "LicenseCountingType": {
        "target": "com.amazonaws.licensemanager#LicenseCountingType",
        "traits": {
          "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
        }
      },
      "DisassociateWhenNotFound": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
```

          "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Status of the license configuration.</p>"
        }
      },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

      }
      },
      "ConsumedLicenseSummaryList": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
        }
      },
      "ManagedResourceSummaryList": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
        }
      },
      "ProductInformationList": {
        "target": "com.amazonaws.licensemanager#ProductInformationList",
        "traits": {
          "smithy.api#documentation": "<p>Product information.</p>"
        }
      },
      "AutomatedDiscoveryInformation": {

        "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
        "traits": {
          "smithy.api#documentation": "<p>Automated discovery information.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n   consumed and enforced by License Manager. Components include specifications for
the license\n   type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n   must be associated with a
host), and the number of licenses purchased and used.</p>"

```
          }
        },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
        "type": "structure",
        "members": {
          "ResourceArn":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of server resource.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
            }
          },
          "AssociationTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
            }
          },
          "AmiAssociationScope": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
        "type": "enum",
        "members":
{
          "AVAILABLE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "AVAILABLE"
            }
          },
          "DISABLED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DISABLED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits":
{
              "smithy.api#documentation": "<p>Type of resource.</p>"
            }
          },
          "ResourceStatus": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Status of the resource.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
            }
          },
```

```
            "AssociationTime": {
              "target": "com.amazonaws.licensemanager#DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
              }
            },

        "ConsumedLicenses": {
              "target": "com.amazonaws.licensemanager#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurations": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfiguration"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
        "members": {
          "UsageOperation": {
            "target": "com.amazonaws.licensemanager#UsageOperation",
            "traits": {
              "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n        </p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {

     "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the conversion task.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "StartTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
```

```
                    "traits": {
                        "smithy.api#documentation":
    "<p>The time the conversion task was started at.</p>"
                    }
                },
                "LicenseConversionTime": {
                    "target": "com.amazonaws.licensemanager#DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
                    }
                },
                "EndTime": {
                    "target": "com.amazonaws.licensemanager#DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
            }
        },
        "com.amazonaws.licensemanager#LicenseConversionTaskId": {
            "type": "string",
            "traits": {

                "smithy.api#length": {
                    "min": 0,
                    "max": 50
                },
                "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
            }
        },
        "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
            "type": "enum",
            "members": {
                "IN_PROGRESS": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "IN_PROGRESS"
                    }
                },
                "SUCCEEDED": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "SUCCEEDED"
                    }
```

```
          },
          "FAILED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED"
            }
          }
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionTasks":
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConversionTask"
      }
    },
    "com.amazonaws.licensemanager#LicenseCountingType": {
      "type": "enum",
      "members": {
        "VCPU": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "vCPU"
          }
        },
        "INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Instance"
          }
        },
        "CORE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Core"
          }
        },
        "SOCKET": {
          "target": "smithy.api#Unit",
           "traits": {
            "smithy.api#enumValue": "Socket"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseDeletionStatus": {
      "type": "enum",
      "members": {
```

```
      "PENDING_DELETE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_DELETE"
        }
      },
      "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DELETED"
        }
      }
    }
  }
},
"com.amazonaws.licensemanager#LicenseList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanager#License"
  }
},
"com.amazonaws.licensemanager#LicenseOperationFailure": {

  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
      }
    },
    "ResourceType": {
      "target": "com.amazonaws.licensemanager#ResourceType",
      "traits": {
        "smithy.api#documentation": "<p>Resource type.</p>"
      }
    },
    "ErrorMessage": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Error message.</p>"
      }
    },
    "FailureTime": {
      "target": "com.amazonaws.licensemanager#DateTime",
      "traits": {

"smithy.api#documentation": "<p>Failure time.</p>"
        }
```

```
        },
        "OperationName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the operation.</p>"
          }
        },
        "ResourceOwnerId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
          }
        },
        "OperationRequestedBy": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
          }
        },
        "MetadataList": {

          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Reserved.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailureList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
      }
    },
    "com.amazonaws.licensemanager#LicenseSpecification": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
```

```
        "smithy.api#required": {}
      }
    },
    "AmiAssociationScope": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
  }
},
"com.amazonaws.licensemanager#LicenseSpecifications": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanager#LicenseSpecification"
  }
},
"com.amazonaws.licensemanager#LicenseStatus": {
  "type": "enum",
  "members": {
    "AVAILABLE": {
      "target": "smithy.api#Unit",

      "traits": {
        "smithy.api#enumValue": "AVAILABLE"
      }
    },
    "PENDING_AVAILABLE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "PENDING_AVAILABLE"
      }
    },
    "DEACTIVATED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "DEACTIVATED"
      }
    },
    "SUSPENDED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "SUSPENDED"
      }
```

```
        },
        "EXPIRED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EXPIRED"
          }
        },

     "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseUsage": {
      "type": "structure",
      "members": {
        "EntitlementUsages": {
          "target": "com.amazonaws.licensemanager#EntitlementUsageList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlement usages.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

      }
    },
    "com.amazonaws.licensemanager#LicenseUsageException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "LicenseUsageFailure",
          "httpResponseCode": 412
```

```
        },
        "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 412
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
      },
      "output": {

    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
```

```
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
          "type": "structure",
          "members": {
            "LicenseConfigurationAssociations": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
              "traits": {
                "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
              }
            },
            "NextToken": {

  "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListDistributedGrants": {
          "type": "operation",
          "input": {
```

```
          "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
        },
      "output": {
          "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
        },
      "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
           },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
      "traits": {
          "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
        "type": "structure",
        "members": {
          "GrantArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {

          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n      <ul>\n         <li>\n            <p>\n               <code>LicenseArn</code>\n            </p>\n
</li>\n         <li>\n            <p>\n               <code>GrantStatus</code>\n            </p>\n         </li>\n
```

```
<li>\n        <p>\n          <code>GranteePrincipalARN</code>\n        </p>\n      </li>\n
<li>\n        <p>\n          <code>ProductSKU</code>\n        </p>\n      </li>\n      <li>\n
<p>\n          <code>LicenseIssuerName</code>\n
</p>\n      </li>\n    </ul>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
          "type": "structure",
          "members": {
            "Grants": {
              "target": "com.amazonaws.licensemanager#GrantList",

      "traits": {
                "smithy.api#documentation": "<p>Distributed grant details.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
        },
        "errors": [
          {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
```

```
      },
      "traits": {
        "smithy.api#input": { }
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
      "type": "structure",
      "members": {
        "LicenseOperationFailureList": {
          "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
          "traits": {
            "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": { }
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
      },

"output": {
        "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
```

```
                {
                  "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                  "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the license configurations
 for your account.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
            "type": "structure",
            "members": {
                "LicenseConfigurationArns": {
                    "target": "com.amazonaws.licensemanager#StringList",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanager#Filters",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n        Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n        Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n
<p>\n        <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n        Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
   </li>\n        <li>\n        <p>\n        <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n        Logical operators are
```

```
          <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n       </ul>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
        "type": "structure",
        "members": {
            "LicenseConfigurations": {
                "target": "com.amazonaws.licensemanager#LicenseConfigurations",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license configurations.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",

                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
        "type": "structure",
        "members": {
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>\n        Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n      </p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTasks": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
            }
          },
```

```json
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target":
"com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
```

```
        }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n        <ul>\n            <li>\n                <p>\n                    <code>LicenseConfigurationArn</code>\n
</p>\n            </li>\n        </ul>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits":
{
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
        "type": "structure",
        "members": {
            "ReportGenerators": {
                "target": "com.amazonaws.licensemanager#ReportGeneratorList",
                "traits": {
                    "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
```

```
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
```

```
                    "traits": {

          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                  },
                  "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                      "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                  }
                }
              },
              "traits": {
                "smithy.api#input": {}
              }
            }
          },
          "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
            "type": "structure",
            "members": {
              "LicenseSpecifications": {
                "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                "traits": {
                  "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
                }
              },
              "NextToken": {
                 "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#ListLicenseVersions": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
```

```
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
```

```
                "target": "com.amazonaws.licensemanager#LicenseList",
                "traits": {
                    "smithy.api#documentation": "<p>License details.</p>"
                }
            },

    "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#ListLicenses": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#ListLicensesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#ListLicensesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
    }
```

        },
      "com.amazonaws.licensemanager#ListLicensesRequest": {
        "type": "structure",
        "members": {
          "LicenseArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
            }
          },

      "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>Beneficiary</code>\n   </p>\n
      \n   \n   <p>\n   <code>ProductSKU</code>\n   </p>\n   \n
   \n   <p>\n   <code>Fingerprint</code>\n   </p>\n   \n   \n
<p>\n   <code>Status</code>\n   </p>\n   \n   "
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token
 for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicensesResponse": {
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#LicenseList",
            "traits": {
              "smithy.api#documentation": "<p>License details.</p>"
            }
          },

```json
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrants": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while specifying the\n        recipient as this Amazon Web Services account, your organization, or an organizational unit\n    (OU) to which this member account belongs.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
            "smithy.api#required": {}
          }
        },
        "Filters": {
```

```
              "target": "com.amazonaws.licensemanager#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
      <ul>\n         <li>\n            <p>\n               <code>ParentArn</code>\n            </p>\n         </li>\n
<li>\n            <p>\n               <code>GranteePrincipalArn</code>\n            </p>\n         </li>\n      </ul>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
          "type": "structure",
          "members": {
            "Grants": {
              "target": "com.amazonaws.licensemanager#GrantList",
              "traits": {
                "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
          "type": "structure",
```

```
        "members": {
          "GrantArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n    <ul>\n      <li>\n        <p>\n          <code>ProductSKU</code>\n        </p>\n
      </li>\n      <li>\n        <p>\n          <code>LicenseIssuerName</code>\n        </p>\n
</li>\n      <li>\n        <p>\n          <code>LicenseArn</code>\n        </p>\n      </li>\n
<li>\n        <p>\n          <code>GrantStatus</code>\n        </p>\n      </li>\n      <li>\n
<p>\n
      <code>GranterAccountId</code>\n        </p>\n      </li>\n    </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
        "type": "structure",
        "members": {
          "Grants": {
            "target": "com.amazonaws.licensemanager#GrantList",
            "traits": {
              "smithy.api#documentation": "<p>Received grant details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicenses": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists received licenses.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {

    "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n           <li>\n              <p>\n                 <code>Beneficiary</code>\n              </p>\n
   </li>\n           <li>\n              <p>\n                 <code>ProductSKU</code>\n
      </p>\n           </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
```

```
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#GrantedLicenseList",
          "traits": {
            "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },

        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n                <code>ProductSKU</code>\n            </p>\n
```

```
            </li>\n          <li>\n          <p>\n          <code>Status</code>\n          </p>\n          </li>\n
<li>\n          <p>\n          <code>Fingerprint</code>\n          </p>\n          </li>\n          <li>\n
<p>\n          <code>IssuerName</code>\n          </p>\n          </li>\n          <li>\n          <p>\n
<code>Beneficiary</code>\n          </p>\n          </li>\n          </ul>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
        "type": "structure",
        "members": {
            "Licenses": {
                "target": "com.amazonaws.licensemanager#GrantedLicenseList",
                "traits": {
                    "smithy.api#documentation": "<p>Received license details.</p>"
                }
            },

            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListResourceInventory": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FailedDependencyException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#InventoryFilterList",
```

```
        "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n        <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
        </li>\n        <li>\n        <p>\n        <code>application_name</code> - The name of the
application. \n        Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
    </li>\n        <li>\n        <p>\n        <code>license_included</code> - The type of license included. \n
    Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n        Possible values
are <code>sql-server-enterprise</code> | \n        <code>sql-server-standard</code> | \n        <code>sql-
server-web</code> | \n        <code>windows-server-datacenter</code>.</p>\n        </li>\n        <li>\n
    <p>\n        <code>platform</code> - The platform of the resource.\n        Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n        </li>\n        <li>\n        <p>\n
    <code>resource_id</code> - The ID of the resource. \n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n        <p>\n
    <code>tag:<key></code> - The key/value combination of a tag assigned \n        to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n        <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n        </li>\n        </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#ListResourceInventoryResponse": {
    "type": "structure",
    "members": {
        "ResourceInventoryList": {
            "target": "com.amazonaws.licensemanager#ResourceInventoryList",
            "traits": {
                "smithy.api#documentation": "<p>Information
about the resources.</p>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#ListTagsForResource": {
    "type": "operation",
```

```json
      "input": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

    "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
```

```
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the tags.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListTokens": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListTokensRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListTokensResponse"
      },
      "errors":
[
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists your tokens.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTokensRequest": {
      "type": "structure",
      "members": {
        "TokenIds": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
```

```
                "smithy.api#documentation": "<p>Token IDs.</p>"
              }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n          <li>\n              <p>\n                  <code>LicenseArns</code>\n              </p>\n          </li>\n
</ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
          "Tokens": {
            "target": "com.amazonaws.licensemanager#TokenList",
            "traits": {
              "smithy.api#documentation": "<p>Received token details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
```

```
        },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
 {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n        consumption details by resource at a selected point in time. Use this action to audit the\n        current
license consumption for any license inventory and configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          },
```

```
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#Filters",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n      are supported:</p>\n      <ul>\n        <li>\n          <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
     Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n
 <li>\n          <p>\n              <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n          <p>\n
  <code>resourceAccount</code> - The ID of the account that owns the resource. \n          Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n      </ul>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {

"LicenseConfigurationUsageList": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
        "traits": {
          "smithy.api#documentation": "<p>Information about the license configurations.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
```

```
        },
        "traits": {
          "smithy.api#output": { }
        }
      },
      "com.amazonaws.licensemanager#Location": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#Long": {
        "type": "long"
      },
      "com.amazonaws.licensemanager#ManagedResourceSummary": {
        "type": "structure",
        "members": {
          "ResourceType":
{
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
            }
          },
          "AssociationCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
        }
      },
      "com.amazonaws.licensemanager#MaxSize100": {

"type": "integer",
        "traits": {
          "smithy.api#range": {
            "min": 1,
            "max": 100
          }
        }
      },
```

```
"com.amazonaws.licensemanager#MaxSize3StringList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 3
        }
    }
},
"com.amazonaws.licensemanager#Message": {
    "type": "string"
},
"com.amazonaws.licensemanager#Metadata": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The key name.</p>"
            }
        },

        "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
    }
},
"com.amazonaws.licensemanager#MetadataList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#Metadata"
    }
},
"com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
```

```
        },
      "traits": {
        "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#Options": {
      "type": "structure",
      "members": {
        "ActivationOverrideBehavior": {
          "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
          "traits": {
            "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n        Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n        <ul>\n          <li>\n                <p>As a license administrator, you can
optionally specify an\n                <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
</li>\n          <li>\n                <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n                account in your
organization.</p>\n          </li>\n          <li>\n                <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n                <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n            are activating the grant.</p>\n          </li>\n        </ul>\n        <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n          <dd>\n                <p>Use this value to activate a grant
without replacing any member accounts\n                active grants for the same product.</p>\n          </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n          <dd>\n                <p>Use this value to activate a
grant and disable other active grants in any\n                member accounts
 for the same product. This action will also replace their\n                previously activated grants with this activated
grant.</p>\n          </dd>\n        </dl>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n        override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#OrganizationConfiguration": {
      "type": "structure",
      "members": {
        "EnableIntegration": {
          "target": "com.amazonaws.licensemanager#Boolean",
          "traits": {
            "smithy.api#default": false,
            "smithy.api#documentation":
```

"<p>Enables Organizations integration.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
    }
},
"com.amazonaws.licensemanager#PrincipalArnList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#Arn"
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 1
        }
    }
},
"com.amazonaws.licensemanager#ProductInformation": {
    "type": "structure",
    "members": {
        "ResourceType": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Resource type. The
possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                "smithy.api#required": {}
            }
        },
        "ProductInformationFilterList": {
            "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
            "traits": {
                "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n   <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n   <p>The following filters and are supported when
the resource type \n   is <code>SSM_MANAGED</code>:</p>\n
        \n   \n   <p>\n   <code>Application Name</code> - The name of the
application.\n   Logical operator is <code>EQUALS</code>.</p>\n   \n   \n
<p>\n   <code>Application Publisher</code> - The publisher of the application.\n   Logical
operator is <code>EQUALS</code>.</p>\n   \n   \n   <p>\n
<code>Application Version</code> - The version of the application.\n   Logical operator is
<code>EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>Platform

Name</code> - The name of the platform.\n   Logical operator is <code>EQUALS</code>.</p>\n
\n   \n   <p>\n   <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
     \n   \n   <p>\n   <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>. The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n   </p>\n   \n   \n   <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n   Logical operator is <code>NOT_EQUALS</code>.</p>\n   \n
\n   <p>\n   <code>License Included</code> - The type of license included.\n   Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n   Possible values are:
 <code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-server-
web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n
<p>The following filters and logical operators are supported when the resource type\n   is
<code>RDS</code>:</p>\n   \n   \n   <p>\n   <code>Engine Edition</code> -
The edition of the database engine.\n   Logical operator is <code>EQUALS</code>.\n   Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n   \n   \n   <p>\n   <code>License Pack</code> - The license
pack.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>data
guard</code> | \n   <code>diagnostic pack sqlt</code> | \n
        <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n   <code>olap</code>.</p>\n
\n   ",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#ProductInformationFilter": {
        "type": "structure",
        "members": {
          "ProductInformationFilterName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Filter name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductInformationFilterValue": {
            "target": "com.amazonaws.licensemanager#StringList",

            "traits": {
              "smithy.api#documentation": "<p>Filter value.</p>"
            }
          },
```

```
        "ProductInformationFilterComparator": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Logical operator.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes product information filters.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformationFilter"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformation"
      }
    },
    "com.amazonaws.licensemanager#ProvisionalConfiguration": {
      "type": "structure",
      "members": {
        "MaxTimeToLiveInMinutes": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#RateLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
```

```
          "aws.protocols#awsQueryError": {
            "code": "RateLimitExceeded",
            "httpResponseCode": 429
          },
          "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 429
        }
    },
    "com.amazonaws.licensemanager#ReceivedMetadata": {
      "type": "structure",
      "members": {
        "ReceivedStatus": {
          "target": "com.amazonaws.licensemanager#ReceivedStatus",
          "traits": {
            "smithy.api#documentation": "<p>Received status.</p>"
          }
        },
        "ReceivedStatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Received
status reason.</p>"
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReceivedStatus": {
      "type": "enum",
      "members": {
        "PENDING_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_WORKFLOW"
          }
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue":
"PENDING_ACCEPT"
      }
    },
    "REJECTED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "REJECTED"
      }
    },
    "ACTIVE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "ACTIVE"
      }
    },
    "FAILED_WORKFLOW": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "FAILED_WORKFLOW"
      }
    },
    "DELETED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "DELETED"
      }
    },
    "DISABLED": {
      "target": "smithy.api#Unit",
       "traits": {
        "smithy.api#enumValue": "DISABLED"
      }
    },
    "WORKFLOW_COMPLETED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "WORKFLOW_COMPLETED"
      }
    }
  }
},
"com.amazonaws.licensemanager#RedirectException": {
  "type": "structure",
  "members": {
    "Location": {
      "target": "com.amazonaws.licensemanager#Location",
```

```json
            "traits": {
                "smithy.api#httpHeader": "Location"
            }
        },
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 308
    }
},
"com.amazonaws.licensemanager#RejectGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#RejectGrantRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#RejectGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
```

```
            }
        },
        "com.amazonaws.licensemanager#RejectGrantRequest": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#RejectGrantResponse": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant ARN.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant version.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#RenewType":
{
            "type": "enum",
            "members": {
```

```
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "WEEKLY": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Weekly"
            }
          },
          "MONTHLY": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Monthly"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#ReportContext": {
        "type": "structure",
        "members": {
          "licenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {

        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
        }
      },
      "com.amazonaws.licensemanager#ReportFrequency": {
        "type": "structure",
        "members": {
          "value": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
              "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n      The only supported value is <code>1</code>.</p>"
            }
          },
          "period": {
```

```
            "target": "com.amazonaws.licensemanager#ReportFrequencyType",

            "traits": {
                "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
    }
},
"com.amazonaws.licensemanager#ReportFrequencyType": {
    "type": "enum",
    "members": {
        "DAY": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "DAY"
            }
        },
        "WEEK": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "WEEK"
            }
        },
        "MONTH": {
            "target": "smithy.api#Unit",
            "traits":
{
                "smithy.api#enumValue": "MONTH"
            }
        }
    }
},
"com.amazonaws.licensemanager#ReportGenerator": {
    "type": "structure",
    "members": {
        "ReportGeneratorName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>"
            }
        },
        "ReportType": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
```

```
                "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
            }
        },
        "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
                "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
            }
        },
        "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits": {
                "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
            }
        },
        "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
            }
        },
        "LastRunStatus": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
            }
        },
        "LastRunFailureReason":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
            }
        },
        "LastReportGenerationTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
            }
        },
        "ReportCreatorAccount": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
            }
        },
```

"Description": {
            "target": "com.amazonaws.licensemanager#String",

    "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
        },
        "S3Location": {
            "target": "com.amazonaws.licensemanager#S3Location",
            "traits": {
                "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
            }
        },
        "CreateTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Time the report was created.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
                "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
            }

}
    },
    "traits": {
        "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
    }
},
"com.amazonaws.licensemanager#ReportGeneratorList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#ReportGenerator"
    }
},
"com.amazonaws.licensemanager#ReportGeneratorName": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        }
    }
},
"com.amazonaws.licensemanager#ReportType": {

```
            "type": "enum",
            "members": {
                "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
                    }
                },
                "LICENSE_CONFIGURATION_USAGE_REPORT": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "LicenseConfigurationUsageReport"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#ReportTypeList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#ReportType"
            }
        },
        "com.amazonaws.licensemanager#ResourceInventory": {
            "type": "structure",
            "members": {
                "ResourceId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the resource.</p>"
                    }
                },
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#ResourceType",
                    "traits":
{
                        "smithy.api#documentation": "<p>Type of resource.</p>"
                    }
                },
                "ResourceArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
                    }
                },
                "Platform": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
```

```
              "smithy.api#documentation": "<p>Platform of the resource.</p>"
            }
          },
          "PlatformVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
            }
          },
          "ResourceOwningAccountId":
{
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details about a resource.</p>"
          }
        },
        "com.amazonaws.licensemanager#ResourceInventoryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#ResourceInventory"
          }
        },
        "com.amazonaws.licensemanager#ResourceLimitExceededException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "ResourceLimitExceeded",

              "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#ResourceNotFoundException": {
          "type": "structure",
          "members": {
```

```
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "InvalidResource.NotFound",
        "httpResponseCode": 400
      },
      "smithy.api#documentation": "<p>The resource cannot be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  }
},
"com.amazonaws.licensemanager#ResourceType": {
  "type": "enum",
  "members": {
    "EC2_INSTANCE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EC2_INSTANCE"
      }
    },
    "EC2_HOST": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EC2_HOST"
      }
    },
    "EC2_AMI": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EC2_AMI"
      }
    },
    "RDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "RDS"
      }
    },
    "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
      }
    }
  }
```

```
      },
      "com.amazonaws.licensemanager#S3Location": {
        "type": "structure",
        "members": {
          "bucket": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
            }
          },
          "keyPrefix": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
        }
      },
      "com.amazonaws.licensemanager#ServerInternalException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InternalError",
            "httpResponseCode": 500
          },
          "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanager#SignedToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 4096
          }
        }
      },
      "com.amazonaws.licensemanager#StatusReasonMessage": {
```

```
            "type": "string",
            "traits":
{
            "smithy.api#length": {
                "min": 0,
                "max": 400
            },
            "smithy.api#pattern": "^[\\s\\S]+$"
        }
    },
    "com.amazonaws.licensemanager#String": {
        "type": "string"
    },
    "com.amazonaws.licensemanager#StringList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#String"
        }
    },
    "com.amazonaws.licensemanager#Tag": {
        "type": "structure",
        "members": {
            "Key": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Tag key.</p>"
                }
            },
            "Value": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation":
"<p>Tag value.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#TagKeyList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#String"
        }
    },
    "com.amazonaws.licensemanager#TagList": {
        "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanager#Tag"
        }
      },
      "com.amazonaws.licensemanager#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#TagResourceResponse"
        },
        "errors": [
          {
            "target":
"com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits":
{
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
```

```
          "traits": {
            "smithy.api#documentation": "<p>One or more tags.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#TokenData": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
        "TokenType": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "ExpirationTime": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
format.</p>"
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Data specified by the caller.</p>"
          }
        },
        "RoleArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
          }
        }
      },
      "traits": {

        "smithy.api#documentation": "<p>Describes a token.</p>"
      }
    },
    "com.amazonaws.licensemanager#TokenList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#TokenData"
      }
    },
    "com.amazonaws.licensemanager#TokenString": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 4096
        },
        "smithy.api#pattern": "^\\S+$"
      }
    },
    "com.amazonaws.licensemanager#TokenType": {
      "type": "enum",
      "members": {
        "REFRESH_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REFRESH_TOKEN"
          }
```

```
      }
    }
  },
  "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
{
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
    }
  },
```

```
"com.amazonaws.licensemanager#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licensemanager#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
    },
    "output": {

"target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
              },
              {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "LicenseConfigurationStatus": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
              "traits": {
                "smithy.api#documentation": "<p>New status of the license configuration.</p>"
              }
            },
            "LicenseRules": {
              "target": "com.amazonaws.licensemanager#StringList",
              "traits": {

   "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
   configuration is licenseAffinityToHost.</p>"
              }
            },
            "LicenseCount": {
              "target": "com.amazonaws.licensemanager#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
              }
            },
            "LicenseCountHardLimit": {
```

```
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>New description of the license configuration.</p>"
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>New product information.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        }
      },
      "traits":
{
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
```

```
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
 {
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
            "smithy.api#required": {}
          }
        },
        "ReportGeneratorName": {
```

```
              "target": "com.amazonaws.licensemanager#ReportGeneratorName",
              "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>",
                "smithy.api#required": {}
              }
            },
            "Type": {
              "target": "com.amazonaws.licensemanager#ReportTypeList",
              "traits": {
                "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n        <ul>\n          <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n          </li>\n          <li>\n          <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n          </li>\n        </ul>",
                "smithy.api#required": {}
              }
            },
            "ReportContext": {
              "target": "com.amazonaws.licensemanager#ReportContext",
              "traits": {
                "smithy.api#documentation": "<p>The report context.</p>",
                "smithy.api#required": {}
              }
            },
            "ReportFrequency": {
              "target": "com.amazonaws.licensemanager#ReportFrequency",
              "traits":
{
                "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                "smithy.api#required": {}
              }
            },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientRequestToken",
              "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
              }
            }
          },
          "traits": {
```

```
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },

                {
                    "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
                },
                {
                    "target": "com.amazonaws.licensemanager#LicenseUsageException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n    <p>You can update the license specifications of AMIs, instances, and
hosts.\n     You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
```

```
            }
          },

      "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "AddLicenseSpecifications": {
                "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                "traits": {
                    "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
                }
            },
            "RemoveLicenseSpecifications": {
                "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                "traits": {
                    "smithy.api#documentation":
"<p>ARNs of the license configurations to remove.</p>"
                }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
      },
      "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
      },
      "com.amazonaws.licensemanager#UpdateServiceSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
          },
          "errors": [
```

```
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
     },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
  }
},
"com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "S3BucketArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where the License
 Manager information is stored.</p>"
      }
    },
    "SnsTopicArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
      }
    },
    "OrganizationConfiguration": {
      "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
      "traits": {
        "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
      }
    },
    "EnableCrossAccountsDiscovery": {
      "target": "com.amazonaws.licensemanager#BoxBoolean",
      "traits": {
```

```
                    "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UsageOperation": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            }
        }
    },
    "com.amazonaws.licensemanager#ValidationException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

    "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    }
    }
}
}
```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
"context"
"fmt"

```
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version
 of the license.
 //
 // This member is required.
 SourceVersion *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus
```

```go
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack,
options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
```

```go
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "DeleteLicense",
	}
}
```

1.20 aws-aws-sdk-go-v2-internal-ini 1.7.1

1.20.1 Available under license :

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil,
 err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType
```

```go
	// The ID of the workspace to associate the license with.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

	// A structure containing data about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
	if err !=
	nil {
		return err
	}
	err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
```

```go
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
  return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack);
  err != nil {
  return err
}
if err = addClientUserAgent(stack, options); err != nil {
  return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
  return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
```

```go
   return err
  }
  if err = addDisableHTTPSMiddleware(stack,
  options); err != nil {
   return err
  }
  return nil
 }


func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
```
AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

name: License Scan

```yaml
on: [pull_request]

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
    matrix:
     python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
        path: sdkbase
        ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
        path: new-ref
        fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
```

```
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)


// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```go
}

	out := result.(*CheckInLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type CheckInLicenseInput struct {

	// License consumption token.
	//
	// This member is required.
	LicenseConsumptionToken *string

	// License beneficiary.
	Beneficiary
	*string

	noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
  return err
  }
  if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
  err != nil {
   return err
  }
  if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
   return err
  }
  if err = addResolveEndpointMiddleware(stack, options); err != nil {
   return err
  }
  if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
   return err
  }
  if err = addRetryMiddlewares(stack, options); err != nil {
   return err
  }
  if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
   return err
  }
  if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
   return err
  }
  if err = addClientUserAgent(stack, options); err != nil {
   return err
  }
  if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
   return err
  }
  if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
   return err
  }
  if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
   return err
  }
  if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
   return
  err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
  middleware.Before); err != nil {
   return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
   return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
```

```go
   return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
   return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
   return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
   return err
  }
  return nil
 }

 func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata {
  return &awsmiddleware.RegisterServiceMetadata{
   Region:       region,
   ServiceID:    ServiceID,
   OperationName: "CheckInLicense",
  }
 }
```

```json
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
          }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
```

```
                "traits": {
                   "smithy.api#documentation": "<p>An operator for filtering results.</p>"
                }
             }
          },
          "traits": {
             "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n     Filters can be used to match a set of resources by specific criteria.</p>"
          }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
          "type": "list",
          "member": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
          }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {

             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
             {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
             },
             {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
             },
             {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
          ],
          "traits": {
             "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
             "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"
             },
             "smithy.api#idempotent": {}
          }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
```

```
{
    "type": "structure",
    "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n     Organizations integration has been enabled, and
which Regions data will be aggregated from.</p>"
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays
the aggregated data for Linux\n     subscriptions.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
        "AmiId": {
```

```
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }

      },
      "AccountID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the instance.</p>"
        }
      },
      "Region": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
        }
      },
      "UsageOperation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
        }
      },
      "ProductCode": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
        "traits": {
```

```
                    "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
                }
            },
            "LastUpdatedTime": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
                }
            },
            "SubscriptionName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],

  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",

        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
```

"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The
AWS region used to dispatch the request.",
            "type": "String"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

            "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ],
            "type": "tree",

```json
"rules": [
  {

    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
              {
                  "ref": "UseFIPS"
              },
                true
            ]
        }
      ],
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
      "type": "error"
  },
  {
      "conditions": [],
      "type": "tree",
      "rules": [
        {

    "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                  {
                      "ref": "UseDualStack"
                  },
                    true
                ]
            }
          ],
          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
          "type": "error"
      },
      {
          "conditions": [],

"endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
```

```
                            }
                          ]
                        }
                      ]
                    },
                    {
                      "conditions": [],
                      "type": "tree",
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "isSet",

               "argv": [
                                {
                                  "ref": "Region"
                                }
                              ]
                            }
                          ],
                          "type": "tree",
                          "rules": [
                            {
                              "conditions": [
                                {
                                  "fn": "aws.partition",
                                  "argv": [
                                    {
                                      "ref": "Region"
                                    }
                                  ],

        "assign": "PartitionResult"
                                }
                              ],
                              "type": "tree",
                              "rules": [
                                {
                                  "conditions": [
                                    {
                                      "fn": "booleanEquals",
                                      "argv": [
                                        {
                                          "ref": "UseFIPS"
                                        },
                                        true
                                      ]
```

```
          },
        {
              "fn": "booleanEquals",
              "argv": [
                {
                   "ref": "UseDualStack"
                },
                true
              ]
          }
        ],
        "type": "tree",
        "rules": [
          {
              "conditions": [
                {

              "fn": "booleanEquals",
                   "argv": [
                     true,
                     {
                        "fn": "getAttr",
                        "argv": [
                          {
                             "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]

}
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  true,
                  {
                     "fn": "getAttr",
                     "argv": [
                       {
                          "ref": "PartitionResult"
                       },

                  "supportsDualStack"
                     ]
                  }
                ]
```

```
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [],
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ]
                  }
                ]
              },
              {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                "type": "error"
              }
            ]
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }
            ],
            "type": "tree",
            "rules": [
              {
```

```
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,

{
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        }
                    ],
                    "type": "tree",

                    "rules": [
                        {
                            "conditions": [],
                            "type": "tree",
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                                        "properties": {},
                                        "headers": {}

                                    },
                                    "type": "endpoint"
                                }
                            ]
                        }
                    ]
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"
                }
            ]
```

        },
                              {
                      "conditions": [
                         {
                            "fn": "booleanEquals",
                            "argv": [
                               {
                                  "ref": "UseDualStack"
                               },
                                true
                            ]
                         }
                      ],
                      "type": "tree",
                      "rules": [
                         {

                      "conditions": [
                            {
                               "fn": "booleanEquals",
                               "argv": [
                                  true,
                                  {
                                     "fn": "getAttr",
                                     "argv": [
                                        {
                                           "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                     ]
                                  }
                               ]
                            }
                         ],
                         "type": "tree",
                         "rules": [
                            {
                               "conditions": [],
                               "type": "tree",
                               "rules": [
                                  {
                                     "conditions": [],

                            "endpoint": {
                                     "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                     "properties": {},
                                     "headers": {}

                                  },
                                  "type": "endpoint"
                               }
                            ]
                         }
                      ]
                   },

         {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                         }
                      ]
                   },
                   {
                      "conditions": [],
                      "type": "tree",
                      "rules": [
                         {
                            "conditions": [],
                            "endpoint": {
                               "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                               "properties": {},
                               "headers": {}
                            },
                            "type": "endpoint"
                         }
                      ]
                   }
                ]
             }
          ]
       },
       {
          "conditions": [],
          "error": "Invalid Configuration: Missing Region",
          "type": "error"
       }
    ]
  }
],
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
       "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
         "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
```

```
              "UseDualStack": false
          }
      },
      {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
              "endpoint": {


"url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
              }
          },
          "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
          }
      },
      {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
              "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
          },
          "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
               "UseDualStack": false
          }
      },
      {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {
              "endpoint": {
                  "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
          },
          "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
          }
      },
      {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
              "endpoint":
```

```
{
                                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                          }
                    },
                    "params": {
                          "Region": "cn-north-1",
                          "UseFIPS": false,
                          "UseDualStack": false
                    }
              },
              {
                    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                          "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                          }
                    },
                    "params": {
                          "Region": "us-gov-east-1",
                          "UseFIPS": true,
                          "UseDualStack": true
                    }
              },
              {
                    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                          "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
                          }
                    },
                    "params": {
                          "Region": "us-gov-east-1",
                          "UseFIPS": true,
                          "UseDualStack": false
                    }
              },
              {
                    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                          "endpoint":
{
                                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
                          }
                    },
                    "params": {
                          "Region": "us-gov-east-1",
                          "UseFIPS": false,
                          "UseDualStack": true
```

```
              }
          },
          {
              "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": false,
                 "UseDualStack": false
              }
          },
          {
              "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
          },
          {
              "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                }
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
          },
          {
              "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "error": "DualStack is enabled but this partition does not support DualStack"
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": false,
                "UseDualStack": true
```

```
                    }
                },
                {

        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
                        }
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": false,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                    },
                    "params": {

                     "Region": "us-isob-east-1",
                        "UseFIPS": true,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                        }
                    },
                    "params": {
                        "Region": "us-isob-east-1",
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
                    "expect": {
                        "error": "DualStack is enabled but this partition does not support DualStack"
                    },
                    "params": {
```

```
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",

  "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack

disabled",

    "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
```

```
                {
                  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                  "expect": {
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false,

         "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                  "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }

        }
            ],
            "version": "1.0"
          }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled LinuxSubscriptionsDiscovery"
          },
          {
            "name": "Disabled",
```

```
                    "value": "Disabled",
                    "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
        "type": "structure",
        "members": {
            "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "OrganizationIntegration": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
                "traits": {
                    "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type":
"operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {

          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Instances"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>AmiID</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>InstanceID</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>AccountID</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>Status</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>Region</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>UsageOperation</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>ProductCode</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>InstanceType</code>\n          </p>\n        </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>contains</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>equals</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>Notequal</code>\n          </p>\n        </li>\n      </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
```

```
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
```

```
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n    organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
            }

          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n     partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n          <code>Subscription</code>\n        </p>\n      </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n
    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n
<li>\n        <p>\n          <code>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n
     <code>Notequal</code>\n        </p>\n      </li>\n    </ul>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
```

```
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                "smithy.api#length": {


                  "min": 1,
                    "max": 16384
                  }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
              },
              {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
```

```
                "documentation": "Not equal operator"
              },
              {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
              }
            ],
            "smithy.api#length": {
              "min": 1,
              "max": 20
            }
          }
        },

        "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
              },
              {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [

{
              "name": "InProgress",
              "value": "InProgress",
              "documentation": "InProgress status"
            },
```

```
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
                    "value": "Successful",
                    "documentation": "Successful status"
                },
                {
                    "name": "Failed",
                    "value": "Failed",
                    "documentation": "Failed status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
```

```
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription.</p>"

          }
          },
          "Type": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
            }
          },
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },

    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required": {}
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>",
```

```
          "smithy.api#required": {}
                }
            },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
               "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n     allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
               "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },

          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
               "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>"
                }
            },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
               "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
               "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
```

                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n   subscriptions.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
          }
        }
      }
    }
}

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
```

```go
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //   - Beneficiary
    //   - ProductSKU
    //   - Fingerprint
    //
    //   - Status
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
```

```
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if
err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
```

```go
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
     Region:       region,
     ServiceID:
       ServiceID,
     OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
```

```go
	if params == nil {
		params = &DisassociateLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*DisassociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type DisassociateLicenseInput struct {

	// The type of license to remove from the workspace.
	//
	//
	This member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to remove the Grafana Enterprise license from.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

	// A structure containing information about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
```

```
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "DisassociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
```

```go
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
{
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "DisassociateLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```go
"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type
 CheckoutLicenseInput struct {

 // Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
```

```
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource
    Name (ARN) of the checkout license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}
```

```go
func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil
	{
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
```

```go
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack);
	err != nil {
		return err
	}
	if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
	middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
```

```go
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
    c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier
    that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see RFC 7518 Digital Signature with
    // RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData
```

```go
	// Amazon Resource Name (ARN) of the license. The license must use the borrow
	// consumption configuration.
	//
	// This member is required.
	LicenseArn *string


	// Information about constraints.
	CheckoutMetadata []types.Metadata


	// Node ID.
	NodeId *string


	noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

	// Information about constraints.
	CheckoutMetadata []types.Metadata


	// Allowed license entitlements.
	EntitlementsAllowed
	[]types.EntitlementData


	// Date and time at which the license checkout expires.
	Expiration *string


	// Date and time at which the license checkout is issued.
	IssuedAt *string


	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string


	// License consumption token.
	LicenseConsumptionToken *string


	// Node ID.
	NodeId *string


	// Signed token.
	SignedToken *string


	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata


	noSmithyDocumentSerde
}
```

```go
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return
```

```go
    err
  }
  if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
  }
  if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
  }
  if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
  }
  if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
    return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
    return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
  }
  return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```go
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide
 to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity
*types.DatetimeRange

// Information about the license.
LicenseMetadata []types.Metadata

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string
```

```go
	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
		return fmt.Errorf("add
protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
```

```
        return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
        return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
    err != nil {
        return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
        return err
    }
    if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
        return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
        return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
        return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
        return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
        return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
        return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
```

```go
    Region:        region,
    ServiceID:     ServiceID,
    OperationName:
"CreateLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version
```

```go
    *string

    noSmithyDocumentSerde
}

type GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return
    err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
```

```go
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
  return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
}
if err = addClientUserAgent(stack, options); err != nil {
  return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
}
if err = addOpGetLicenseValidationMiddleware(stack);
err != nil {
  return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
  return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
  return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
}
return nil
```

```
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "GetLicense",
 }
}
```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
```

```go
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
	if params == nil {
		params = &ListReceivedLicensesInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*ListReceivedLicensesOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type ListReceivedLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//
	//   - ProductSKU
	//   - Status
	//   - Fingerprint
	//   - IssuerName
	//   - Beneficiary
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

	// Received license details.
	Licenses []types.GrantedLicense
```

```go
	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
```

```go
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack);
	err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
	middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
	string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "ListReceivedLicenses",
```

```
    }
}
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership
of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial
revisions, annotations, elaborations, or other modifications
   represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
 incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
```

"<p>Details about an Active Directory identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n   <note>\n   <p>Your estimated bill for charges on the number of users and related
costs will take 48\n   hours to appear for billing periods that haven't closed (marked as Pending billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
        monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p>\n   </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n    resource.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
```

subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"

```json
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/DisassociateUser"
        },
        "smithy.api#idempotent": {}
      }

    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},

"com.amazonaws.licensemanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        },
        "Operation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
```

```json
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter name and value pair that is used
 to return more specific results from a describe\n     operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n     attributes, or IDs.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
```

```
            "traits": {
                "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n      settings such as the subnets to provision VPC endpoints.</p>",
                "smithy.api#required":
{}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
```

```
              }
            },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status
of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          }

        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
```

```
          "smithy.api#required": {}
        }
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits":
{
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
```

```
            }
          }
    },
        "traits": {
          "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",

    "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
```

```
                  "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
              },
              {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
              }

      },
              {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
              },
              {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
              },
              {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
              },
              {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
              }
          ],
          "traits": {
              "aws.api#service": {
                  "sdkId": "License Manager User Subscriptions",
                  "arnNamespace": "license-manager-user-subscriptions",
                  "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
              },
              "aws.auth#sigv4": {
                  "name": "license-manager-user-subscriptions"
              },
              "aws.protocols#restJson1": {},
              "smithy.api#cors": {
                  "additionalAllowedHeaders": [
                      "*",
                      "content-type",
                      "x-amz-content-sha256",
                      "x-amz-user-agent",
                      "x-amzn-platform-id",
                      "x-amzn-trace-id",
                      "content-length",
                      "x-api-key",
                      "authorization",
                      "x-amz-date",
                      "x-amz-security-token",
                      "Access-Control-Allow-Headers",
                      "Access-Control-Allow-Methods",
                      "Access-Control-Allow-Origin"
                  ],
                  "additionalExposedHeaders": [
                      "x-amzn-errortype",
```

```
                    "x-amzn-requestid",
                    "x-amzn-trace-id"


]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n        a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
                "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
                    "documentation": "The AWS region used to dispatch the request.",
                    "type": "String"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack
 endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                    "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }

            },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
```

```
                    "ref": "Endpoint"
                  }
                ]
              }
            ],
            "type": "tree",
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },

                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],

  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
```

```
                                                  "type": "endpoint"
                                            }
                                    ]
                            },
                            {
                                "conditions": [
                                    {
                                        "fn": "isSet",
                                        "argv": [

            {
                                                "ref": "Region"
                                            }
                                    ]
                                }
                            ],
                            "type": "tree",
                            "rules": [
                                {
                                    "conditions": [
                                        {
                                            "fn": "aws.partition",
                                            "argv": [
                                                {
                                                    "ref": "Region"
                                                }
                                            ],
                                            "assign": "PartitionResult"
                                        }
                                    ],
                                    "type": "tree",

"rules": [
                                        {
                                            "conditions": [
                                                {
                                                    "fn": "booleanEquals",
                                                    "argv": [
                                                        {
                                                            "ref": "UseFIPS"
                                                        },
                                                        true
                                                    ]
                                                },
                                                {
                                                    "fn": "booleanEquals",
                                                    "argv": [
                                                        {
```

```
                              "ref": "UseDualStack"

                      },
                      true
                  ]
              }
          ],
          "type": "tree",
          "rules": [
              {
                  "conditions": [
                      {
                          "fn": "booleanEquals",
                          "argv": [
                              true,
                              {
                                  "fn": "getAttr",

"argv": [

                                      {
                                          "ref": "PartitionResult"
                                      },
                                      "supportsFIPS"
                                  ]
                              }
                          ]
                      },
                      {
                          "fn": "booleanEquals",
                          "argv": [
                              true,
                              {

"fn": "getAttr",

                                  "argv": [
                                      {
                                          "ref": "PartitionResult"
                                      },
                                      "supportsDualStack"
                                  ]
                              }
                          ]
                      }
                  ],
                  "type": "tree",
                  "rules": [
                      {
```

```
            "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                              },
                              "type": "endpoint"
                            }
                          ]
                        },
                        {
                          "conditions": [],
                          "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                           "type": "error"
                        }
                      ]
                    },
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseFIPS"
                            },
                            true
                          ]
                        }
                      ],
                      "type": "tree",

        "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
```

```
                            ]
                          }
                        ]
                      }
                    ],
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}

          },
                          "type": "endpoint"
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"
                  }
                ]
              },
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
```

```
                            "fn": "getAttr",
                              "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                              ]
                          }
                        ]
                      }
                    ],
                    "type": "tree",
                    "rules": [
                      {

                          "conditions": [],
                          "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                          },
                          "type": "endpoint"
                        }
                      ]
                    },
                    {
                      "conditions": [],
                      "error": "DualStack is enabled but this partition does
 not support DualStack",
                      "type": "error"
                    }
                  ]
                },
                {
                  "conditions": [],
                  "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }
              ]
            }
          ]
```

```
            },
            {
              "conditions":
[],
              "error": "Invalid Configuration: Missing Region",
              "type": "error"
            }
          ]
        },
        "smithy.rules#endpointTests": {
          "testCases": [
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                }
              },

          "params": {
                "Region": "us-east-1",
```

```
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with
 FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
        },

    "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },

"params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
```

```
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
              }
            },

    "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
            "expect":
{
              "endpoint": {
```

            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack":

true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but

this partition does not support one or both"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",

```
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
          }
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }

},
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
          }
        },
         "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
          "endpoint": {
            "url": "https://example.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false,
          "Endpoint": "https://example.com"
```

```
            }
          },
          {

"documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
             "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "Missing region",
          "expect": {
            "error": "Invalid Configuration: Missing Region"
          }
        }
      ],
      "version": "1.0"
```

```
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/ListIdentityProviders"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "IdentityProviderSummaries"
        }
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
          {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type":
"structure",
        "members": {
            "InstanceSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListProductSubscriptions"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {

    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n      more sets of key-value pairs that you specify.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
          },

      "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceUserSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
```

```
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {

          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }

          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for a user.</p>",
              "smithy.api#required": {}
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          },
          "SubscriptionStartDate": {
```

```
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
                    }
                },
                "SubscriptionEndDate": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                },
                {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/RegisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n      subnets to provision VPC endpoints.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
        "type": "string",

  "traits": {
          "smithy.api#length": {
            "min": 5,
            "max": 200
          },
          "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
```

```
"members": {
  "Subnets": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
      "smithy.api#length": {
        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n      your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n      subnets to provision VPC endpoints, and the security group ID that
 is associated with the VPC\n      endpoints. The security group should permit inbound TCP port 1688
communication from resources\n      in the VPC.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related costs
will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
 For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n        monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/StartProductSubscription"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
```

```
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the user.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required":
{}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            }
```

```
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation":
 "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StringList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {

              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
```

```
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity\n      provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
```

```
        "UpdateSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n      update any combination of settings in a single operation such as the:</p>\n
<ul>\n        <li>\n            <p>Subnets which you want to add to provision VPC endpoints.</p>\n        </li>\n
    <li>\n            <p>Subnets which you want to remove the VPC endpoints from.</p>\n        </li>\n
<li>\n            <p>Security group ID which permits traffic to the VPC endpoints.</p>\n        </li>\n      </ul>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n      require connectivity to activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
          }
        },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID
```

that allows inbound TCP port 1688 communication between resources in\n   your VPC and the VPC endpoints for activation servers.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as\n   the subnets to provision VPC endpoints.</p>"
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
            "type": "structure",
            "members": {
              "message": {
                "target": "smithy.api#String"
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>A parameter is not valid.</p>",
              "smithy.api#error": "client"
            }
          }
        }
      }
    }
    {
      "smithy": "2.0",
      "metadata": {
        "suppressions": [
          {
            "id": "HttpMethodSemantics",
            "namespace": "*"
          },
          {
            "id": "HttpResponseCodeSemantics",
            "namespace": "*"
          },
          {
            "id": "PaginatedTrait",
            "namespace": "*"
          },
          {
            "id": "HttpHeaderTrait",
            "namespace": "*"
          },
          {
            "id": "HttpUriConflict",
            "namespace": "*"
          },

```json
        {
            "id": "Service",
            "namespace": "*"
        }
    ]
},
"shapes": {
    "com.amazonaws.licensemanager#AWSLicenseManager": {
        "type": "service",
        "version": "2018-08-01",
        "operations": [
            {
                "target": "com.amazonaws.licensemanager#AcceptGrant"
            },


            {

                "target": "com.amazonaws.licensemanager#CheckInLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CheckoutLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateGrant"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateGrantVersion"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateToken"
            },
```

```
    {
        "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
        "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
        "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
        "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    }

},
    {
        "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
    {
        "target": "com.amazonaws.licensemanager#GetGrant"
    },
    {
        "target": "com.amazonaws.licensemanager#GetLicense"
    },
    {
        "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
    },
    {
        "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
    },
    {
        "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
    },
    {
        "target": "com.amazonaws.licensemanager#GetLicenseUsage"
    },
    {
        "target": "com.amazonaws.licensemanager#GetServiceSettings"
    },
    {
        "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
    },
    {
        "target": "com.amazonaws.licensemanager#ListDistributedGrants"
```

    },
    {
      "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
    },
    {

      "target": "com.amazonaws.licensemanager#ListLicenseVersions"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListResourceInventory"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTokens"
    },
    {
      "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
    },
    {

```
        "target": "com.amazonaws.licensemanager#RejectGrant"
      },
      {
        "target": "com.amazonaws.licensemanager#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UntagResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager",
        "arnNamespace": "license-manager",
        "cloudFormationName": "LicenseManager",
        "cloudTrailEventSource": "licensemanager.amazonaws.com",
        "endpointPrefix": "license-manager"
      },
      "aws.auth#sigv4": {
        "name": "license-manager"
      },
      "aws.protocols#awsJson1_1": {},
      "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n        Amazon Web Services accounts and on-premises servers.</p>",
      "smithy.api#title": "AWS License Manager",
      "smithy.api#xmlNamespace": {
        "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
      },
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
          "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
```

```
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
                    "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }
            },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]

                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },
                                        true
```

```
                    ]
                }
            ],
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
            "type": "error"
        },
        {

"conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
]
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ],
    "type": "tree",
    "rules": [
```

```json
                                {
                                    "conditions": [
                                        {
                                            "fn": "aws.partition",
                                            "argv": [
                                                {
                                                    "ref": "Region"
                                                }
                                            ],
                                            "assign": "PartitionResult"
                                        }
                                    ],
                                    "type": "tree",
                                    "rules": [
                                        {
                                            "conditions": [
                                                {
                                                    "fn": "booleanEquals",
                                                    "argv": [
                                                        {
                                                            "ref":
"UseFIPS"
                                                        },
                                                        true
                                                    ]
                                                },
                                                {
                                                    "fn": "booleanEquals",
                                                    "argv": [
                                                        {
                                                            "ref": "UseDualStack"
                                                        },
                                                        true
                                                    ]
                                                }
                                            ],
                                            "type": "tree",
                                            "rules": [
                                                {
                                                    "conditions": [
                                                        {
                                                            "fn": "booleanEquals",
                                                            "argv": [
                                                                true,
                                                                {
                                                                    "fn": "getAttr",
                                                                    "argv": [
                                                                        {
```

                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                            }
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },

    "supportsDualStack"

                                ]
                            }
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}

                        },
                        "type": "endpoint"
                    }
                ]
            },
            {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
            }
        ]
    },

```
{
   "conditions": [
      {
         "fn": "booleanEquals",

         "argv": [
            {
               "ref": "UseFIPS"
            },
            true
         ]
      }
   ],
   "type": "tree",
   "rules": [
      {
         "conditions": [
            {
               "fn": "booleanEquals",
               "argv": [
                  true,

                  {
                     "fn": "getAttr",
                     "argv": [
                        {
                           "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                     ]
                  }
               ]
            }
         ],
         "type": "tree",
         "rules": [

            {
               "conditions": [],
               "endpoint": {
                  "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": {},
                  "headers": {}
               },
               "type": "endpoint"
            }
         ]
      },
```

```
                          {
                              "conditions": [],
                              "error": "FIPS is enabled but this
partition does not support FIPS",
                              "type": "error"
                          }
                      ]
                  },
                  {
                      "conditions": [
                          {
                              "fn": "booleanEquals",
                              "argv": [
                                  {
                                      "ref": "UseDualStack"
                                  },
                                  true
                              ]
                          }
                      ],
                      "type": "tree",
                      "rules": [
                          {
                              "conditions": [
                                  {
                                      "fn": "booleanEquals",
                                      "argv": [
                                          true,
                                          {
                                              "fn": "getAttr",
                                              "argv": [
                                                  {
                                                      "ref": "PartitionResult"
                                                  },

"supportsDualStack"

                                              ]
                                          }
                                      ]
                                  }
                              ],
                              "type": "tree",
                              "rules": [
                                  {
                                      "conditions": [],
                                      "endpoint": {
                                          "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
```

```
                                    "properties": {},
                                    "headers": {}

                        },
                              "type": "endpoint"
                          }
                      ]
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ]
              },
              {
                  "conditions": [],
                  "endpoint": {
                      "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                      "properties": {},
                      "headers": {}
                  },
                  "type": "endpoint"
              }
            ]
          }
        ]
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
  ]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
        "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint":
{
              "url": "https://license-manager.af-south-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "af-south-1",
          "UseFIPS": false,
```

```
          "UseDualStack": false
        }
    },
    {
        "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
    },
    {
        "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-northeast-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
    },
    {
        "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-northeast-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
    },
    {
        "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-3.amazonaws.com"
          }
```

```
          },
          "params": {
            "Region": "ap-northeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }

      },
        {
          "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-1.amazonaws.com"

    }
          },
          "params": {
            "Region": "ap-southeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
```

```json
        {
            "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.ap-southeast-3.amazonaws.com"
                }
            },
            "params": {
                "Region": "ap-southeast-3",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.ca-central-1.amazonaws.com"
                }
            },

            "params": {
                "Region": "ca-central-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.eu-central-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "eu-central-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.eu-north-1.amazonaws.com"
                }
```

```
          },
          "params": {
            "Region": "eu-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region":
"eu-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-west-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
          "expect": {

            "endpoint": {
              "url": "https://license-manager.eu-west-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-west-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
```

```
{
  "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.eu-west-3.amazonaws.com"
    }
  },
  "params": {
    "Region": "eu-west-3",
    "UseFIPS": false,

    "UseDualStack": false
  }
},
{
  "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.me-south-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "me-south-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.sa-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "sa-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.us-east-1.amazonaws.com"
    }
  },
  "params": {
```

```
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }

    },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-east-2.amazonaws.com"
              }

          },
            "params": {
              "Region": "us-east-2",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-east-2.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-2",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region
```

```
us-west-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.us-west-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-west-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.us-west-1.amazonaws.com"
                }
              },
              "params": {

        "Region": "us-west-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.us-west-2.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-west-2",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

"expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.us-west-2.amazonaws.com"
                }
              },
              "params": {
```

```
                    "Region": "us-west-2",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,

                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
```

```
        "expect": {
          "endpoint": {
            "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-northwest-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
```

```
                  "UseDualStack": true
            }
      },

{
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                  "endpoint": {
                        "url": "https://license-manager.us-gov-east-1.amazonaws.com"
                  }
            },
            "params": {
                  "Region": "us-gov-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
            }
      },
      {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                  "endpoint": {
                        "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
                  }
            },
            "params": {
                  "Region": "us-gov-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
            }
      },
      {
            "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
            "expect": {
                  "endpoint": {
                        "url": "https://license-manager.us-gov-west-1.amazonaws.com"
                  }
            },
            "params": {
                  "Region": "us-gov-west-1",
                  "UseFIPS": false,
                  "UseDualStack": false
            }
      },
      {
            "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
            "expect": {
                  "endpoint": {
```

```
            "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-west-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-gov-east-1.api.aws"
          }
        },

  "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": false
```

```
              }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
                    }
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled
but this partition does not support DualStack"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": false,
```

```
          "UseDualStack": false
        }
      },

    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
          }
        },

    "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

"expect": {
          "endpoint": {
            "url": "https://example.com"
          }
        },
```

```
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,

      "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error":
  "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
```

```
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanager#AcceptGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

},

    "output": {
        "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"

}
    ],
    "traits": {
        "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
    }
},
"com.amazonaws.licensemanager#AcceptGrantRequest": {
```

```
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant
ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#AccessDeniedException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
```

```json
        "traits": {
            "aws.protocols#awsQueryError": {

"code": "ServiceAccessDenied",
                "httpResponseCode": 401
            },
            "smithy.api#documentation": "<p>Access to resource denied.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 401
        }
    },
    "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
        "type": "enum",
        "members": {
            "DISTRIBUTED_GRANTS_ONLY": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
                }
            },
            "ALL_GRANTS_PERMITTED_BY_ISSUER": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#AllowedOperation": {

"type": "enum",
        "members": {
            "CREATE_GRANT": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CreateGrant"
                }
            },
            "CHECKOUT_LICENSE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CheckoutLicense"
                }
            },
            "CHECKOUT_BORROW_LICENSE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CheckoutBorrowLicense"
```

```
        }
      },
      "CHECK_IN_LICENSE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "CheckInLicense"
        }
      },
      "EXTEND_CONSUMPTION_LICENSE":
{

        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "ExtendConsumptionLicense"
        }
      },
      "LIST_PURCHASED_LICENSES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "ListPurchasedLicenses"
        }
      },
      "CREATE_TOKEN": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "CreateToken"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#AllowedOperationList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#AllowedOperation"
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max":
7
      }
    }
  },
  "com.amazonaws.licensemanager#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 2048
```

```
        },
        "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-
z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
      }
    },
    "com.amazonaws.licensemanager#ArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
      }
    },
    "com.amazonaws.licensemanager#AuthorizationException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError":
{
          "code": "AuthorizationFailure",
          "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n          policy associated with this account.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 403
      }
    },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
      "type": "structure",
      "members": {
        "LastRunTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes automated discovery.</p>"
      }
    },

  "com.amazonaws.licensemanager#Boolean": {
      "type": "boolean",
      "traits": {
```

```
          "smithy.api#default": false
        }
      },
      "com.amazonaws.licensemanager#BorrowConfiguration": {
        "type": "structure",
        "members": {
          "AllowEarlyCheckIn": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxTimeToLiveInMinutes": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
              "smithy.api#required": {}
            }
          }

        },
          "traits": {
            "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
          }
      },
      "com.amazonaws.licensemanager#BoxBoolean": {
        "type": "boolean"
      },
      "com.amazonaws.licensemanager#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanager#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanager#CheckInLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
```

```
              "target": "com.amazonaws.licensemanager#AuthorizationException"

          },
            {
              "target": "com.amazonaws.licensemanager#ConflictException"
          },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
            {
              "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
        }
    },
    "com.amazonaws.licensemanager#CheckInLicenseRequest":
{
        "type": "structure",
        "members": {
          "LicenseConsumptionToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License consumption token.</p>",
                "smithy.api#required": {}
              }
          },
          "Beneficiary": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License beneficiary.</p>"
              }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

```
      },
      "com.amazonaws.licensemanager#CheckInLicenseResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
          },
          {
            "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
      "type": "structure",
      "members":
{
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
            "smithy.api#required": {}
          }
        },
        "DigitalSignatureMethod": {
          "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
          "traits": {
            "smithy.api#documentation": "<p>Digital signature method. The possible
 value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
            "smithy.api#required": {}
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
```

            },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientToken",
              "traits": {
                "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },
        "EntitlementsAllowed":
{
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "SignedToken": {
          "target": "com.amazonaws.licensemanager#SignedToken",
          "traits": {
            "smithy.api#documentation": "<p>Signed token.</p>"
          }
        },

```json
        "IssuedAt": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
          }
        },
        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
```

```
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target":
"com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n           <p>If the
account that created the license is the same that is performing the check out, you must\n          specify the account
as the beneficiary.</p>\n        </note>"
    }
  },
  "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
    "type": "structure",
    "members": {
      "ProductSKU": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product
SKU.</p>",
          "smithy.api#required": {}
        }
      },
      "CheckoutType": {
        "target": "com.amazonaws.licensemanager#CheckoutType",
        "traits": {
          "smithy.api#documentation": "<p>Checkout type.</p>",
          "smithy.api#required": {}
        }
      },
      "KeyFingerprint": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
          "smithy.api#required": {}
        }
      },
```

```json
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        }
      },
      "traits": {

    "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
      "type": "structure",
      "members": {
        "CheckoutType": {
          "target": "com.amazonaws.licensemanager#CheckoutType",
          "traits": {
            "smithy.api#documentation": "<p>Checkout type.</p>"
          }
        },
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
```

```
        },
        "EntitlementsAllowed": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
          }
        },
        "SignedToken": {
          "target": "com.amazonaws.licensemanager#SignedToken",
          "traits": {
            "smithy.api#documentation": "<p>Signed token.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "IssuedAt": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
          }
        },
        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutType": {
      "type": "enum",
      "members": {
        "PROVISIONAL": {
          "target": "smithy.api#Unit",
```

```
            "traits": {
                "smithy.api#enumValue": "PROVISIONAL"
            }
        },
        "PERPETUAL": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "PERPETUAL"
            }
        }
    }
},
"com.amazonaws.licensemanager#ClientRequestToken": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 36
        }
    }
},
"com.amazonaws.licensemanager#ClientToken": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 2048
        },
        "smithy.api#pattern": "^\\S+$"
    }
},
"com.amazonaws.licensemanager#ConflictException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
    },
    "traits": {
        "aws.protocols#awsQueryError": {
            "code": "ConflictException",
            "httpResponseCode":
409
        },
        "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 409
```

```
          }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
          "type": "structure",
          "members": {
            "ResourceType": {
              "target": "com.amazonaws.licensemanager#ResourceType",
              "traits": {
                "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
              }
            },
            "ConsumedLicenses": {
              "target": "com.amazonaws.licensemanager#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
              }
            }
          },

          "traits": {
            "smithy.api#documentation": "<p>Details about license consumption.</p>"
          }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
          }
        },
        "com.amazonaws.licensemanager#ConsumptionConfiguration": {
          "type": "structure",
          "members": {
            "RenewType": {
              "target": "com.amazonaws.licensemanager#RenewType",
              "traits": {
                "smithy.api#documentation": "<p>Renewal frequency.</p>"
              }
            },
            "ProvisionalConfiguration": {
              "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
              "traits": {
                "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
              }
            },
            "BorrowConfiguration": {
              "target": "com.amazonaws.licensemanager#BorrowConfiguration",
              "traits": {
                "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateGrant": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateGrantRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateGrantResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target":
"com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateGrantRequest": {
```

```
      "type": "structure",
      "members": {
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant name.</p>",
            "smithy.api#required": {}

  }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Principals": {
          "target": "com.amazonaws.licensemanager#PrincipalArnList",
          "traits": {
            "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n        (ARN):</p>\n    <ul>\n        <li>\n          <p>An Amazon Web Services
account, which includes only the account specified.</p>\n        </li>\n      </ul>\n      <ul>\n        <li>\n
    <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n        </li>\n      </ul>\n
<ul>\n
        <li>\n          <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n      </ul>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
```

```
                "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },

        "com.amazonaws.licensemanager#CreateGrantResponse": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Grant ARN.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanager#GrantStatus",
              "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#CreateGrantVersion":
        {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
    }
  },
  "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
    "type": "structure",
    "members": {
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientToken",
        "traits": {
          "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

  "smithy.api#required": {}
        }
      },
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
```

```
            "smithy.api#documentation": "<p>Grant name.</p>"
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "StatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Grant status reason.</p>"
          }
        },
        "SourceVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the grant.</p>"
          }
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
            "smithy.api#documentation": "<p>The options specified for the grant.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
},
"com.amazonaws.licensemanager#CreateGrantVersionResponse": {
  "type": "structure",
  "members": {
    "GrantArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Grant ARN.</p>"
      }
    },
    "Status": {
```

```
              "target": "com.amazonaws.licensemanager#GrantStatus",
              "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>New version of the grant.</p>"
              }
            }
          },
          "traits":
{
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#CreateLicense": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#RedirectException"
             },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
```

```
      "traits": {
         "smithy.api#documentation": "<p>Creates a license.</p>"
      }
   },
   "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
      "type": "operation",
      "input": {
         "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
      },
      "output": {
         "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
      },
      "errors": [
         {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
         },
         {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
         },
         {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
         {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
         },
         {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
         },
         {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
         }
      ],
      "traits": {
         "smithy.api#documentation": "<p>Creates a license configuration.</p>\n         <p>A license configuration
is an abstraction of a customer license agreement that can be\n         consumed and enforced by License Manager.
Components include specifications for the license\n         type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n         Dedicated Instance, Dedicated Host, or
all of these), license affinity  to host (how long a \n         license must be associated with a host), and the number of
licenses purchased and used.</p>"
      }
   },
   "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
         "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
               "smithy.api#documentation": "<p>Name of the license configuration.</p>",
```

```
                    "smithy.api#required": {}
                  }
                },
                "Description": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Description of the license configuration.</p>"
                  }
                },
                "LicenseCountingType": {
                  "target": "com.amazonaws.licensemanager#LicenseCountingType",
                  "traits": {
                    "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
                    "smithy.api#required": {}
                  }
                },
                "LicenseCount": {
                  "target": "com.amazonaws.licensemanager#BoxLong",
                  "traits": {
                    "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
                  }
                },
                "LicenseCountHardLimit": {
                  "target": "com.amazonaws.licensemanager#BoxBoolean",
                  "traits": {
                    "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n        blocks the launch of new instances.</p>"
                  }
                },
                "LicenseRules": {
                  "target":
 "com.amazonaws.licensemanager#StringList",
                  "traits": {
                    "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n        vary by dimension, as follows.</p>\n
<ul>\n        <li>\n          <p>\n            <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
      <code>licenseAffinityToHost</code> |\n            <code>maximumCores</code> |
<code>minimumCores</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n            <code>maximumCores</code> |
<code>minimumCores</code> |\n          <code>maximumSockets</code> | <code>minimumSockets</code> |\n
      <code>maximumVcpus</code> | <code>minimumVcpus</code>\n          </p>\n        </li>\n
<li>\n
  <p>\n            <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n            <code>maximumSockets</code> |
<code>minimumSockets</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n            <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n          </p>\n        </li>\n        </ul>\n        <p>The unit for
```

<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for <code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n <code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n <code>True</code> and <code>False</code>.</p>"
```
                }
            },

    "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                    "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
                }
            },
            "DisassociateWhenNotFound": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            },
            "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                    "smithy.api#documentation": "<p>Product information.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
        "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {

          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
            "smithy.api#required": {}
          }
        },
        "SourceLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n      For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
```

the <i>License Manager User Guide</i>.</p>",
                    "smithy.api#required": {}
                }
            },
            "DestinationLicenseContext":
{
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI in
the <i>License Manager User Guide</i>.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

            "traits": {
                    "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {

```
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a report generator.</p>"
    }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
    "type": "structure",
    "members": {
        "ReportGeneratorName":
{
            "target": "com.amazonaws.licensemanager#ReportGeneratorName",
            "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>",
                "smithy.api#required": {}
            }
        },
        "Type": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
                "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n       <ul>\n          <li>\n             <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n          </li>\n          <li>\n             <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n          </li>\n
</ul>",
                "smithy.api#required": {}

    }
        },
```

          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits": {
              "smithy.api#documentation": "<p>Frequency by which reports are generated. Reports can be
generated daily, monthly, or weekly.</p>",
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientRequestToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique,
 case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

```
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product
name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product SKU.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits": {
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region for the license.</p>",
              "smithy.api#required": {}
```

```
    }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
              "smithy.api#required": {}
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>",
              "smithy.api#required": {}
            }
          },
          "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license.</p>"
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          }
        },
```

```
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersion": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target":
"com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>",
            "smithy.api#required": {}
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#Issuer",
```

```
        "traits":
{
            "smithy.api#documentation": "<p>License issuer.</p>",
            "smithy.api#required": {}
        }
    },
    "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
            "smithy.api#documentation": "<p>Home Region of the license.</p>",
            "smithy.api#required": {}
        }
    },
    "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
            "smithy.api#required": {}
        }
    },
    "LicenseMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
            "smithy.api#documentation": "<p>Information about the license.</p>"
        }
    },
    "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",
            "smithy.api#required": {}
        }
    },
    "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
            "smithy.api#required": {}
        }
    },

    "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
                "smithy.api#documentation": "<p>License status.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the license.</p>"
            }
          }
        },
        "traits": {

  "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New version of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
```

```
{}
      }
    },
    "com.amazonaws.licensemanager#CreateToken": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateTokenRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateTokenResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },

        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateTokenRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
```

```
            "target": "com.amazonaws.licensemanager#Arn",
            "traits":
{
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
                "smithy.api#required": {}
            }
        },
        "RoleArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n        License Manager does not check whether the roles are in use.</p>"
            }
        },
        "ExpirationInDays": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
                "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
            }
        },
        "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
            "traits": {
                "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n        to the amr claim of the JWT token.</p>"
            }
        },
        "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
                "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#CreateTokenResponse": {
    "type": "structure",
    "members": {
        "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
```

```
              "traits": {
                  "smithy.api#documentation": "<p>Token ID.</p>"
              }
          },
          "TokenType": {
            "target": "com.amazonaws.licensemanager#TokenType",
            "traits": {
              "smithy.api#documentation": "<p>Token type.</p>"
            }
          },
          "Token": {
            "target": "com.amazonaws.licensemanager#TokenString",
            "traits": {
              "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
            }
          }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DateTime": {
      "type": "timestamp"
    },
    "com.amazonaws.licensemanager#DatetimeRange": {
      "type": "structure",
      "members": {
        "Begin": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Start of the time range.</p>",
              "smithy.api#required": {}
            }
        },
        "End": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>End of the time range.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteGrant": {
      "type": "operation",
      "input": {
```

```
          "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
         },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
        }
      }
    },
    "com.amazonaws.licensemanager#DeleteGrantRequest": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
          }
```

```
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Current version of the
grant.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteGrantResponse": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation":
"<p>Grant version.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteLicense": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
```

```
      },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },

    },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
    },

  "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
```

```
              {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
            }
          },
          "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
            "type": "structure",
            "members":
  {
              "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>ID of the license configuration.</p>",
                  "smithy.api#required": {}
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
            "type": "structure",
            "members": {},
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
            },
            "errors": [
              {
```

```
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {

  "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
```

```
            "type": "structure",
            "members": { },
            "traits": {
                "smithy.api#output": { }
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                        "smithy.api#required": { }
                    }
                },
                "SourceVersion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Current version of the license.</p>",
                        "smithy.api#required":
{ }
                    }
                }
            },
            "traits": {
                "smithy.api#input": { }
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseResponse": {
            "type": "structure",
            "members": {
                "Status": {
                    "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>License status.</p>"
                    }
                },
                "DeletionDate": {
                    "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": { }
            }
```

```
      },
      "com.amazonaws.licensemanager#DeleteToken": {
        "type": "operation",
         "input": {
          "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },

      {

            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteTokenRequest": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>",
              "smithy.api#required": {}
            }
          }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteTokenResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}


       }
      },
      "com.amazonaws.licensemanager#DigitalSignatureMethod": {
        "type": "enum",
        "members": {
          "JWT_PS384": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "JWT_PS384"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#Entitlement": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement name.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
            }
          },
          "MaxCount": {
            "target": "com.amazonaws.licensemanager#Long",
            "traits": {
              "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
            }
          },
          "Overage": {
```

```json
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement unit.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowCheckIn": {
            "target":
"com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#EntitlementData": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement data name.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement
data value.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement data unit.</p>",
              "smithy.api#required": {}
            }
          }
```

        },
      "traits": {
        "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementDataList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementData"
      }
    },
    "com.amazonaws.licensemanager#EntitlementDataUnit": {
      "type": "enum",
      "members": {
        "COUNT": {
          "target": "smithy.api#Unit",
          "traits": {

"smithy.api#enumValue": "Count"
          }
        },
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "SECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Seconds"
          }
        },
        "MICROSECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Microseconds"
          }
        },
        "MILLISECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Milliseconds"
          }
        },
        "BYTES": {
          "target": "smithy.api#Unit",

```
        "traits": {
          "smithy.api#enumValue": "Bytes"
        }
      },
      "KILOBYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes"
        }
      },
      "MEGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes"
        }
      },
      "GIGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes"
        }
      },
      "TERABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabytes"
        }
      },
      "BITS":
{
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits"
        }
      },
      "KILOBITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits"
        }
      },
      "MEGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabits"
        }
      },
      "GIGABITS": {
```

```
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits"
          }
        },
        "TERABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits"

}
        },
        "PERCENT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Percent"
          }
        },
        "BYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bytes/Second"
          }
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
           "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabytes/Second"
          }
        },
```

```
      "BITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits/Second"
        }
      },
      "KILOBITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits/Second"
        }
      },
      "MEGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue":
"Megabits/Second"
        }
      },
      "GIGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits/Second"
        }
      },
      "TERABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits/Second"
        }
      },
      "COUNT_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Count/Second"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#EntitlementList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Entitlement"
    }
  },
  "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
    "type": "structure",
```

```
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#EntitlementUnit": {
        "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "SECONDS":
{
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Seconds"
            }
          },
          "MICROSECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Microseconds"
            }
          },
          "MILLISECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Milliseconds"
            }
          },
          "BYTES": {
            "target": "smithy.api#Unit",
            "traits": {
```

```
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits"
      }
    },
    "KILOBITS": {
      "target": "smithy.api#Unit",
      "traits": {

        "smithy.api#enumValue": "Kilobits"
      }
    },
    "MEGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits"
      }
    },
    "GIGABITS": {
      "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue": "Gigabits"
      }
    },
    "TERABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabits"
      }
    },
    "PERCENT": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Percent"
      }
    },
    "BYTES_PER_SECOND": {
      "target":
"smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes/Second"
      }
    },
    "KILOBYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes/Second"
      }
    },
    "MEGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes/Second"
      }
    },
    "GIGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes/Second"
      }
    },
    "TERABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {

"smithy.api#enumValue": "Terabytes/Second"
      }
    },
```

```
      "BITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits/Second"
        }
      },
      "KILOBITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits/Second"
        }
      },
      "MEGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabits/Second"
        }
      },
      "GIGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits/Second"
        }
      },

  "TERABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits/Second"
        }
      },
      "COUNT_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Count/Second"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#EntitlementUsage": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement usage name.</p>",
          "smithy.api#required": {}
        }
```

```
        },
        "ConsumedValue": {
          "target": "com.amazonaws.licensemanager#String",

        "traits": {
            "smithy.api#documentation": "<p>Resource usage consumed.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList":
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementUsage"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>",
            "smithy.api#required": {}

        }
        },
        "DryRun": {
          "target": "com.amazonaws.licensemanager#Boolean",
          "traits": {
            "smithy.api#default": false,
            "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
      "type": "structure",
      "members": {
```

```
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },

        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#FailedDependencyException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        },
        "ErrorCode": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "FailedDependency",
          "httpResponseCode": 424
        },
        "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 424
      }
    },
    "com.amazonaws.licensemanager#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#FilterName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
          }
        },
```

```
        "Values": {
          "target": "com.amazonaws.licensemanager#FilterValues",
          "traits": {
            "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n      describe operation. Filters can be used
to match a set of resources by specific criteria,\n      such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanager#FilterLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "FilterLimitExceeded",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Filter"
      }
    },
    "com.amazonaws.licensemanager#FilterName": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#FilterValue": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#FilterValues": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#FilterValue",
        "traits": {
```

```
                    "smithy.api#xmlName": "item"
                }
            }
        },
        "com.amazonaws.licensemanager#Filters": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#Filter",
                "traits": {
                    "smithy.api#xmlName": "item"
                }
            }
        },
        "com.amazonaws.licensemanager#GetAccessToken": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
 tokens\n        are valid for one hour.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetAccessTokenRequest": {
            "type": "structure",
            "members": {
                "Token": {
```

```
              "target": "com.amazonaws.licensemanager#TokenString",
              "traits": {
                "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
            "traits": {
              "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
        "type": "structure",
        "members": {
          "AccessToken": {
            "target": "com.amazonaws.licensemanager#TokenString",
            "traits": {
              "smithy.api#documentation": "<p>Temporary access token.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetGrant": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetGrantRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetGrantResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
```

            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetGrantRequest": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetGrantResponse": {
          "type": "structure",
          "members": {
            "Grant": {
              "target": "com.amazonaws.licensemanager#Grant",
              "traits": {

```
            "smithy.api#documentation": "<p>Grant details.</p>"
          }
        }
      },
      "traits": {

"smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#GetLicense": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetLicenseRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetLicenseResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"

      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#GetLicenseConfiguration": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
    },
    "output": {
```

```
                   "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
               },
               "errors": [
                   {
                       "target": "com.amazonaws.licensemanager#AccessDeniedException"
                   },
                   {
                       "target": "com.amazonaws.licensemanager#AuthorizationException"
                   },
                   {
                       "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                   },
                   {
                       "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                   },
                   {
                       "target": "com.amazonaws.licensemanager#ServerInternalException"
                   }
               ],
               "traits": {
                   "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
               }
           },
           "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
               "type": "structure",
               "members": {
                   "LicenseConfigurationArn": {
                       "target": "com.amazonaws.licensemanager#String",
                       "traits": {
                           "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                           "smithy.api#required": {}
                       }
                   }
               },
               "traits":
{
                   "smithy.api#input": {}
               }
           },
           "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
               "type": "structure",
               "members": {
                   "LicenseConfigurationId": {
                       "target": "com.amazonaws.licensemanager#String",
                       "traits": {
                           "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
```

```
      }
    },
    "LicenseConfigurationArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
      }
    },
    "Name": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Name of the license configuration.</p>"
      }
    },
    "Description": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Description of the license configuration.</p>"
      }
    },
    "LicenseCountingType": {
      "target": "com.amazonaws.licensemanager#LicenseCountingType",
      "traits": {
        "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
      }
    },
    "LicenseRules": {
      "target": "com.amazonaws.licensemanager#StringList",
      "traits": {
        "smithy.api#documentation": "<p>License rules.</p>"
      }
    },
    "LicenseCount": {
      "target": "com.amazonaws.licensemanager#BoxLong",

      "traits": {
        "smithy.api#documentation": "<p>Number of available licenses.</p>"
      }
    },
    "LicenseCountHardLimit": {
      "target": "com.amazonaws.licensemanager#BoxBoolean",
      "traits": {
        "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
      }
    },
    "ConsumedLicenses": {
      "target": "com.amazonaws.licensemanager#BoxLong",
      "traits": {
```

```
        "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>License configuration status.</p>"
      }

    },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
        }
      },
      "ConsumedLicenseSummaryList": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
        }
      },
      "ManagedResourceSummaryList": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanager#TagList",
        "traits": {
          "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
        }
      },
      "ProductInformationList": {
        "target": "com.amazonaws.licensemanager#ProductInformationList",
        "traits": {
          "smithy.api#documentation": "<p>Product information.</p>"
        }
      },
      "AutomatedDiscoveryInformation": {
        "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
        "traits": {
          "smithy.api#documentation": "<p>Automated discovery information.</p>"
        }
      },
      "DisassociateWhenNotFound": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
```

```
            "traits": {
                "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#GetLicenseConversionTask": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {

    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
    }
},
"com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
    "type": "structure",
    "members": {
        "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
                "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
```

```
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
            }
```

```
                },

    "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                }
            },
            "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
            "type": "structure",
            "members": {
                "LicenseManagerReportGeneratorArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
                        "smithy.api#required":
{}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
            "type": "structure",
            "members": {
                "ReportGenerator": {
                    "target": "com.amazonaws.licensemanager#ReportGenerator",
                    "traits": {
                        "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
```

```
    },
    "com.amazonaws.licensemanager#GetLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseResponse": {
      "type": "structure",
      "members": {
        "License": {
          "target": "com.amazonaws.licensemanager#License",
          "traits": {
            "smithy.api#documentation": "<p>License details.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },

  "com.amazonaws.licensemanager#GetLicenseUsage": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
      },
      "errors": [
        {
```

```json
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {

  "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
      "type": "structure",
      "members": {
        "LicenseUsage": {
          "target":
"com.amazonaws.licensemanager#LicenseUsage",
          "traits": {
            "smithy.api#documentation": "<p>License usage details.</p>"
          }
        }
```

```
      },
      "traits": {
        "smithy.api#output": { }
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"
```

```
          }
        },
        "SnsTopicArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
          }
        },
        "OrganizationConfiguration": {
          "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n        cross-account discovery.</p>"
          }
        },
        "EnableCrossAccountsDiscovery": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
          }

      },
        "LicenseManagerResourceShareArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n        provides member accounts with access to this share.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Grant": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
```

```
      "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant name.</p>",
                "smithy.api#required": {}
              }
            },
            "ParentArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Parent ARN.</p>",
                "smithy.api#required": {}
              }
            },
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>License ARN.</p>",
                "smithy.api#required": {}
              }
            },
            "GranteePrincipalArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {

  "smithy.api#documentation": "<p>The grantee principal ARN.</p>",
                "smithy.api#required": {}
              }
            },
            "HomeRegion": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                "smithy.api#required": {}
              }
            },
            "GrantStatus": {
              "target": "com.amazonaws.licensemanager#GrantStatus",
              "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>",
                "smithy.api#required": {}
              }
            },
            "StatusReason": {
              "target": "com.amazonaws.licensemanager#StatusReasonMessage",
              "traits": {
                "smithy.api#documentation": "<p>Grant status reason.</p>"
              }
            },
```

```
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant version.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
        "traits": {
          "smithy.api#documentation": "<p>Granted operations.</p>",
          "smithy.api#required": {}
        }
      },
      "Options": {
        "target": "com.amazonaws.licensemanager#Options",
        "traits": {
          "smithy.api#documentation": "<p>The options specified for the grant.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>Describes a grant.</p>"
    }
  },
  "com.amazonaws.licensemanager#GrantList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Grant"
    }
  },
  "com.amazonaws.licensemanager#GrantStatus": {
    "type": "enum",
    "members": {
      "PENDING_WORKFLOW": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_WORKFLOW"
        }
      },
      "PENDING_ACCEPT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_ACCEPT"
        }
      },
      "REJECTED": {
```

```
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"


        }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits":
{
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
      "type": "structure",
      "members": {
```

```
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
      }
    },
    "LicenseName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>License name.</p>"
      }

},
    "ProductName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product name.</p>"
      }
    },
    "ProductSKU": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product SKU.</p>"
      }
    },
    "Issuer": {
      "target": "com.amazonaws.licensemanager#IssuerDetails",
      "traits": {
        "smithy.api#documentation": "<p>Granted license issuer.</p>"
      }
    },
    "HomeRegion": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanager#LicenseStatus",
      "traits": {
        "smithy.api#documentation": "<p>Granted license status.</p>"
      }
    },
    "Validity": {
      "target": "com.amazonaws.licensemanager#DatetimeRange",
      "traits": {
        "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
```

ISO8601-UTC format.</p>"
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>"
            }
          },
          "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Granted license metadata.</p>"
            }
          },
          "CreateTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

  }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Version of the granted license.</p>"
            }
          },
          "ReceivedMetadata": {
            "target": "com.amazonaws.licensemanager#ReceivedMetadata",
            "traits": {
              "smithy.api#documentation": "<p>Granted license received metadata.</p>"
            }
          }
        },
        "traits": {

```
          "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
        }
      },
      "com.amazonaws.licensemanager#GrantedLicenseList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#GrantedLicense"
        }
      },
      "com.amazonaws.licensemanager#ISO8601DateTime":
 {
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-
1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
          }
        },
        "com.amazonaws.licensemanager#Integer": {
          "type": "integer"
        },
        "com.amazonaws.licensemanager#InvalidParameterValueException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "InvalidParameterValueProvided",
              "httpResponseCode": 400
            },
            "smithy.api#documentation":
"<p>One or more parameter values are not valid.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#InvalidResourceStateException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
```

```
        },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InvalidResourceState",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },

    "com.amazonaws.licensemanager#InventoryFilter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Condition": {
          "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
          "traits": {
            "smithy.api#documentation": "<p>Condition of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      }

    },
      "traits": {
        "smithy.api#documentation": "<p>An inventory filter.</p>"
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterCondition": {
      "type": "enum",
      "members": {
        "EQUALS": {
          "target": "smithy.api#Unit",
```

```
            "traits": {
              "smithy.api#enumValue": "EQUALS"
            }
          },
          "NOT_EQUALS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "NOT_EQUALS"
            }
          },
          "BEGINS_WITH": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "BEGINS_WITH"
            }
          },
          "CONTAINS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CONTAINS"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#InventoryFilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#InventoryFilter"
        }
      },
      "com.amazonaws.licensemanager#Issuer": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>",
              "smithy.api#required": {}
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
            }
          }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#IssuerDetails": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>"
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#License": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

      "smithy.api#documentation": "<p>License name.</p>"
            }
```

```
      },
      "ProductName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product name.</p>"
        }
      },
      "ProductSKU": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product SKU.</p>"
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#IssuerDetails",
        "traits": {
          "smithy.api#documentation": "<p>License issuer.</p>"
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {

"smithy.api#documentation": "<p>Home Region of the license.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
        "traits": {
          "smithy.api#documentation": "<p>License status.</p>"
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License beneficiary.</p>"
        }
      },
      "Entitlements": {
```

```
                "target": "com.amazonaws.licensemanager#EntitlementList",
                    "traits": {
                        "smithy.api#documentation": "<p>License entitlements.</p>"
                    }
                },
                "ConsumptionConfiguration": {
                    "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
                    }
                },
                "LicenseMetadata": {
                    "target": "com.amazonaws.licensemanager#MetadataList",
                    "traits": {
                        "smithy.api#documentation": "<p>License metadata.</p>"
                    }
                },
                "CreateTime": {
                    "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>License
creation time.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License version.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
            }
        },
        "com.amazonaws.licensemanager#LicenseConfiguration": {
            "type": "structure",
            "members": {
                "LicenseConfigurationId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
                    }
                },
                "LicenseConfigurationArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
```

```
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules.</p>"
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
```

```
            "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the license configuration.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

          }
        },
        "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
          }
        },
        "ManagedResourceSummaryList": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>Product information.</p>"
          }
        },
        "AutomatedDiscoveryInformation": {

          "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
          "traits": {
            "smithy.api#documentation": "<p>Automated discovery information.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n          consumed and enforced by License Manager. Components include specifications for
the license\n          type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n          must be associated with a
host), and the number of licenses purchased and used.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
        "type": "structure",
        "members": {
          "ResourceArn":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of server resource.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
            }
          },
          "AssociationTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
            }
          },
          "AmiAssociationScope": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
        "type": "enum",
        "members":
{
          "AVAILABLE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "AVAILABLE"
            }
          },
          "DISABLED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DISABLED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits":
{
              "smithy.api#documentation": "<p>Type of resource.</p>"
            }
          },
          "ResourceStatus": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Status of the resource.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
            }
          },
```

```
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
          }
        },

    "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurations": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfiguration"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
      "members": {
        "UsageOperation": {
          "target": "com.amazonaws.licensemanager#UsageOperation",
          "traits": {
            "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n          </p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {

      "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the conversion task.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "StartTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
```

```
            "traits": {
                "smithy.api#documentation":
"<p>The time the conversion task was started at.</p>"
            }
        },
        "LicenseConversionTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
                "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
            }
        },
        "EndTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
                "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
    }
},
"com.amazonaws.licensemanager#LicenseConversionTaskId": {
    "type": "string",
    "traits": {

        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
    }
},
"com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
    "type": "enum",
    "members": {
        "IN_PROGRESS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "IN_PROGRESS"
            }
        },
        "SUCCEEDED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "SUCCEEDED"
            }
```

            },
            "FAILED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "FAILED"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#LicenseConversionTasks":
{
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTask"
          }
        },
        "com.amazonaws.licensemanager#LicenseCountingType": {
          "type": "enum",
          "members": {
            "VCPU": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "vCPU"
              }
            },
            "INSTANCE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "Instance"
              }
            },
            "CORE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "Core"
              }
            },
            "SOCKET": {
              "target": "smithy.api#Unit",
               "traits": {
                "smithy.api#enumValue": "Socket"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#LicenseDeletionStatus": {
          "type": "enum",
          "members": {

```json
      "PENDING_DELETE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_DELETE"
        }
      },
      "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DELETED"
        }
      }
    }
},
"com.amazonaws.licensemanager#LicenseList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanager#License"
  }
},
"com.amazonaws.licensemanager#LicenseOperationFailure": {

  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
      }
    },
    "ResourceType": {
      "target": "com.amazonaws.licensemanager#ResourceType",
      "traits": {
        "smithy.api#documentation": "<p>Resource type.</p>"
      }
    },
    "ErrorMessage": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Error message.</p>"
      }
    },
    "FailureTime": {
      "target": "com.amazonaws.licensemanager#DateTime",
      "traits": {

"smithy.api#documentation": "<p>Failure time.</p>"
      }
```

```
        },
        "OperationName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the operation.</p>"
          }
        },
        "ResourceOwnerId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
          }
        },
        "OperationRequestedBy": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
          }
        },
        "MetadataList": {

          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Reserved.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailureList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
      }
    },
    "com.amazonaws.licensemanager#LicenseSpecification": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
```

```
      "smithy.api#required": {}
      }
    },
    "AmiAssociationScope": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
  }
},
"com.amazonaws.licensemanager#LicenseSpecifications": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanager#LicenseSpecification"
  }
},
"com.amazonaws.licensemanager#LicenseStatus": {
  "type": "enum",
  "members": {
    "AVAILABLE": {
      "target": "smithy.api#Unit",

      "traits": {
        "smithy.api#enumValue": "AVAILABLE"
      }
    },
    "PENDING_AVAILABLE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "PENDING_AVAILABLE"
      }
    },
    "DEACTIVATED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "DEACTIVATED"
      }
    },
    "SUSPENDED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "SUSPENDED"
      }
```

```
        },
        "EXPIRED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EXPIRED"
          }
        },

      "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseUsage": {
      "type": "structure",
      "members": {
        "EntitlementUsages": {
          "target": "com.amazonaws.licensemanager#EntitlementUsageList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlement usages.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

      }
    },
    "com.amazonaws.licensemanager#LicenseUsageException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "LicenseUsageFailure",
          "httpResponseCode": 412
```

        },
        "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource launch.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 412
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
      },
      "output": {

    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the resource associations for the specified license configuration.</p>\n   <p>Resource associations need not consume licenses from a license configuration. \n   For example, an AMI or a stopped instance might not consume a license (depending on \n the license rules).</p>"
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",

```
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
          "type": "structure",
          "members": {
            "LicenseConfigurationAssociations": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
              "traits": {
                "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
              }
            },
            "NextToken": {

"target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListDistributedGrants": {
          "type": "operation",
          "input": {
```

"target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
    },
  "output": {
    "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
    },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
     },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
  }
},
"com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
  "type": "structure",
  "members": {
    "GrantArns": {
      "target": "com.amazonaws.licensemanager#ArnList",
      "traits": {

      "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanager#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>LicenseArn</code>\n   </p>\n
\n   \n   <p>\n   <code>GrantStatus</code>\n   </p>\n   \n

```
                    <li>\n          <p>\n              <code>GranteePrincipalARN</code>\n          </p>\n        </li>\n
<li>\n          <p>\n              <code>ProductSKU</code>\n          </p>\n        </li>\n        <li>\n
<p>\n              <code>LicenseIssuerName</code>\n
                    </p>\n        </li>\n        </ul>"
                  }
                },
                "NextToken": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                  }
                },
                "MaxResults": {
                  "target": "com.amazonaws.licensemanager#MaxSize100",
                  "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                  }
                }
              },
              "traits": {
                "smithy.api#input": {}
              }
            },
            "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
              "type": "structure",
              "members": {
                "Grants": {
                  "target": "com.amazonaws.licensemanager#GrantList",

            "traits": {
                        "smithy.api#documentation": "<p>Distributed grant details.</p>"
                  }
                },
                "NextToken": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                  }
                }
              },
              "traits": {
                "smithy.api#output": {}
              }
            },
            "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
              "type": "operation",
              "input": {
                "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
```

        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
        },
        "errors": [
          {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
          "type": "structure",
          "members": {
            "LicenseOperationFailureList": {
              "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
              "traits": {
                "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
        },

    "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
```

```
              {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Lists the license configurations
 for your account.</p>"
            }
          },
        "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
            "type": "structure",
            "members": {
              "LicenseConfigurationArns": {
                "target": "com.amazonaws.licensemanager#StringList",
                "traits": {
                  "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
                  "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
              },
              "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
                }
              },
              "Filters": {
                "target": "com.amazonaws.licensemanager#Filters",
                "traits": {
                  "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n          <li>\n            <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n          Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n          Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n          <li>\n
<p>\n            <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n          Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
    </li>\n          <li>\n          <p>\n            <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n          Logical operators are
```

```
        <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n      </ul>"
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          }
        },
        "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
          "type": "structure",
          "members": {
            "LicenseConfigurations": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurations",
              "traits": {
                "smithy.api#documentation": "<p>Information about the license configurations.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",

              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
```

```
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
          "type": "structure",
          "members": {
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {

          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#Filters",
              "traits": {
                "smithy.api#documentation": "<p>\n      Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n    </p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
          "type": "structure",
          "members": {
            "LicenseConversionTasks": {
              "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
              "traits": {
                "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
              }
            },
```

```
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target":
"com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n        <ul>\n          <li>\n            <p>\n              <code>LicenseConfigurationArn</code>\n
</p>\n          </li>\n        </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits":
{

              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
        "type": "structure",
        "members": {
          "ReportGenerators": {
            "target": "com.amazonaws.licensemanager#ReportGeneratorList",
            "traits": {
              "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
```

```
        }
      },
      "traits": {
        "smithy.api#output": { }
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
            "smithy.api#required": { }
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
```

```
            "traits": {

      "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
        "type": "structure",
        "members": {
          "LicenseSpecifications": {
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
              "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseVersions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
```

```
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
    "type": "structure",
    "members": {
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#MaxSize100",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
    "type": "structure",
    "members": {
      "Licenses": {
```

```
            "target": "com.amazonaws.licensemanager#LicenseList",
            "traits": {
                "smithy.api#documentation": "<p>License details.</p>"
            }
        },

"NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#ListLicenses": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#ListLicensesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#ListLicensesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
    }
```

```
    },
    "com.amazonaws.licensemanager#ListLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },

      "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n    <ul>\n        <li>\n          <p>\n              <code>Beneficiary</code>\n        </p>\n
   </li>\n       <li>\n         <p>\n             <code>ProductSKU</code>\n         </p>\n        </li>\n
 <li>\n          <p>\n             <code>Fingerprint</code>\n        </p>\n       </li>\n       <li>\n
<p>\n            <code>Status</code>\n         </p>\n       </li>\n     </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token
for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicensesResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#LicenseList",
          "traits": {
            "smithy.api#documentation": "<p>License details.</p>"
          }
        },
```

```
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {

      "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListReceivedGrants": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
             },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n        recipient as this Amazon Web Services account, your organization, or an organizational unit\n
   (OU) to which this member account belongs.</p>"
          }
```

```
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
            "smithy.api#required": {}
          }
        },
        "Filters": {
```

```
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
    <ul>\n        <li>\n          <p>\n            <code>ParentArn</code>\n          </p>\n        </li>\n
<li>\n          <p>\n            <code>GranteePrincipalArn</code>\n          </p>\n        </li>\n      </ul>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
        "type": "structure",
        "members": {
            "Grants": {
                "target": "com.amazonaws.licensemanager#GrantList",
                "traits": {
                    "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
        "type": "structure",
```

```
        "members": {
          "GrantArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
             "traits": {
               "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
               "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n    <ul>\n       <li>\n         <p>\n            <code>ProductSKU</code>\n         </p>\n
    </li>\n       <li>\n         <p>\n            <code>LicenseIssuerName</code>\n         </p>\n
</li>\n       <li>\n         <p>\n            <code>LicenseArn</code>\n         </p>\n       </li>\n
<li>\n         <p>\n            <code>GrantStatus</code>\n         </p>\n       </li>\n       <li>\n
<p>\n
      <code>GranterAccountId</code>\n         </p>\n       </li>\n    </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
               "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
               "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    }
  },
  "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
    "type": "structure",
    "members": {
      "Grants": {
        "target": "com.amazonaws.licensemanager#GrantList",
         "traits": {
           "smithy.api#documentation": "<p>Received grant details.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
```

```
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicenses": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists received licenses.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
```

          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {

      "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanager#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>Beneficiary</code>\n   </p>\n
      \n   \n   <p>\n   <code>ProductSKU</code>\n
      </p>\n   \n   "
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"

```
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#GrantedLicenseList",
          "traits": {
            "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },

        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>ProductSKU</code>\n            </p>\n
```

```
    </li>\n    <li>\n    <p>\n    <code>Status</code>\n    </p>\n    </li>\n
<li>\n    <p>\n    <code>Fingerprint</code>\n    </p>\n    </li>\n    <li>\n
<p>\n    <code>IssuerName</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Beneficiary</code>\n    </p>\n    </li>\n    </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
            "type": "structure",
            "members": {
                "Licenses": {
                    "target": "com.amazonaws.licensemanager#GrantedLicenseList",
                    "traits": {
                        "smithy.api#documentation": "<p>Received license details.</p>"
                    }
                },

                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListResourceInventory": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FailedDependencyException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#InventoryFilterList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n            <p>\n                <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
        </li>\n        <li>\n            <p>\n                <code>application_name</code> - The name of the
application. \n            Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
    </li>\n        <li>\n            <p>\n                <code>license_included</code> - The type of license included. \n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n            Possible values
are <code>sql-server-enterprise</code> | \n            <code>sql-server-standard</code> | \n            <code>sql-
server-web</code> | \n            <code>windows-server-datacenter</code>.</p>\n        </li>\n        <li>\n
    <p>\n                <code>platform</code> - The platform of the resource.\n            Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n        </li>\n        <li>\n            <p>\n
    <code>resource_id</code> - The ID of the resource. \n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n            <p>\n
    <code>tag:<key></code> - The key/value combination of a tag assigned \n            to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n            <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n        </li>\n        </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    }
},
"com.amazonaws.licensemanager#ListResourceInventoryResponse": {
    "type": "structure",
    "members": {
        "ResourceInventoryList": {
            "target": "com.amazonaws.licensemanager#ResourceInventoryList",
            "traits": {
                "smithy.api#documentation": "<p>Information
about the resources.</p>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#ListTagsForResource": {
    "type": "operation",
```

```
      "input": {
         "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
      },
      "output": {
         "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
      },
      "errors": [
         {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
         },
         {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
         },
         {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
         {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
         },
         {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
         }
      ],
      "traits": {
         "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
      }
   },
   "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
         "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
               "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

      "smithy.api#required": {}
            }
         }
      },
      "traits": {
         "smithy.api#input": {}
      }
   },
   "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
         "Tags": {
```

```
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the tags.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#ListTokens": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListTokensRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListTokensResponse"
    },
    "errors":
[
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists your tokens.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListTokensRequest": {
    "type": "structure",
    "members": {
      "TokenIds": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
```

```
            "smithy.api#documentation": "<p>Token IDs.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n        <li>\n          <p>\n            <code>LicenseArns</code>\n          </p>\n        </li>\n
</ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListTokensResponse": {
      "type": "structure",
      "members": {
        "Tokens": {
          "target": "com.amazonaws.licensemanager#TokenList",
          "traits": {
            "smithy.api#documentation": "<p>Received token details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
```

        },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
 {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
             {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n   consumption details by resource at a selected point in time. Use this action to audit the\n   current
license consumption for any license inventory and configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
              }
            },

```
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#Filters",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n          <li>\n            <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
       Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n
<li>\n            <p>\n                <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n          <li>\n            <p>\n
   <code>resourceAccount</code> - The ID of the account that owns the resource. \n          Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n        </ul>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {

"LicenseConfigurationUsageList": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
```

```
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Location": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Long": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#ManagedResourceSummary": {
      "type": "structure",
      "members": {
        "ResourceType":
{
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
          }
        },
        "AssociationCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
      }
    },
    "com.amazonaws.licensemanager#MaxSize100": {

      "type": "integer",
      "traits": {
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
```

```
"com.amazonaws.licensemanager#MaxSize3StringList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 3
        }
    }
},
"com.amazonaws.licensemanager#Message": {
    "type": "string"
},
"com.amazonaws.licensemanager#Metadata": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The key name.</p>"
            }
        },

        "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
    }
},
"com.amazonaws.licensemanager#MetadataList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#Metadata"
    }
},
"com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#Options": {
        "type": "structure",
        "members": {
          "ActivationOverrideBehavior": {
            "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
            "traits": {
              "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n         Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n         <ul>\n            <li>\n                <p>As a license administrator, you can
optionally specify an\n               <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
</li>\n            <li>\n                <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n               account in your
organization.</p>\n            </li>\n            <li>\n                <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n               <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n            are activating the grant.</p>\n            </li>\n         </ul>\n         <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n            <dd>\n                <p>Use this value to activate a grant
without replacing any member accounts\n               active grants for the same product.</p>\n            </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n            <dd>\n                <p>Use this value to activate a
grant and disable other active grants in any\n               member accounts
 for the same product. This action will also replace their\n               previously activated grants with this activated
grant.</p>\n            </dd>\n         </dl>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n         override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
      },
      "com.amazonaws.licensemanager#OrganizationConfiguration": {
        "type": "structure",
        "members": {
          "EnableIntegration": {
            "target": "com.amazonaws.licensemanager#Boolean",
            "traits": {
              "smithy.api#default": false,
              "smithy.api#documentation":
```

```
        "<p>Enables Organizations integration.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
    }
},
"com.amazonaws.licensemanager#PrincipalArnList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#Arn"
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 1
        }
    }
},
"com.amazonaws.licensemanager#ProductInformation": {
    "type": "structure",
    "members": {
        "ResourceType": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Resource type. The
possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                "smithy.api#required": {}
            }
        },
        "ProductInformationFilterList": {
            "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
            "traits": {
                "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n          <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n          <p>The following filters and are supported when
the resource type \n          is <code>SSM_MANAGED</code>:</p>\n
        <ul>\n          <li>\n                <p>\n                    <code>Application Name</code> - The name of the
application.\n             Logical operator is <code>EQUALS</code>.</p>\n             </li>\n          <li>\n
<p>\n                <code>Application Publisher</code> - The publisher of the application.\n             Logical
operator is <code>EQUALS</code>.</p>\n             </li>\n          <li>\n                <p>\n
<code>Application Version</code> - The version of the application.\n             Logical operator is
<code>EQUALS</code>.</p>\n             </li>\n          <li>\n                <p>\n                    <code>Platform
```

Name</code> - The name of the platform.\n   Logical operator is <code>EQUALS</code>.</p>\n
\n   \n   <p>\n   <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n

     \n   \n   <p>\n   <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>. The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n   </p>\n   \n   \n   <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n   Logical operator is <code>NOT_EQUALS</code>.</p>\n   \n
\n   <p>\n   <code>License Included</code> - The type of license included.\n   Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n   Possible values are:
 <code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-server-
web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n
<p>The following filters and logical operators are supported when the resource type\n   is
<code>RDS</code>:</p>\n   \n   \n   <p>\n   <code>Engine Edition</code> -
The edition of the database engine.\n   Logical operator is <code>EQUALS</code>.\n   Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n   \n   \n   <p>\n   <code>License Pack</code> - The license
pack.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>data
guard</code> | \n   <code>diagnostic pack sqlt</code> | \n

          <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n   <code>olap</code>.</p>\n
\n   ",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#ProductInformationFilter": {
        "type": "structure",
        "members": {
            "ProductInformationFilterName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Filter name.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductInformationFilterValue": {
                "target": "com.amazonaws.licensemanager#StringList",

            "traits": {
                    "smithy.api#documentation": "<p>Filter value.</p>"
                }
            },
```

```
        "ProductInformationFilterComparator": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Logical operator.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes product information filters.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformationFilter"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformation"
      }
    },
    "com.amazonaws.licensemanager#ProvisionalConfiguration": {
      "type": "structure",
      "members": {
        "MaxTimeToLiveInMinutes": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#RateLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
```

```
        "aws.protocols#awsQueryError": {
          "code": "RateLimitExceeded",
          "httpResponseCode": 429
        },
        "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 429
      }
    },
    "com.amazonaws.licensemanager#ReceivedMetadata": {
      "type": "structure",
      "members": {
        "ReceivedStatus": {
          "target": "com.amazonaws.licensemanager#ReceivedStatus",
          "traits": {
            "smithy.api#documentation": "<p>Received status.</p>"
          }
        },
        "ReceivedStatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Received
status reason.</p>"
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReceivedStatus": {
      "type": "enum",
      "members": {
        "PENDING_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_WORKFLOW"
          }
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
```

```
        "traits": {
            "smithy.api#enumValue":
"PENDING_ACCEPT"
        }
    },
    "REJECTED": {
        "target": "smithy.api#Unit",
        "traits": {
            "smithy.api#enumValue": "REJECTED"
        }
    },
    "ACTIVE": {
        "target": "smithy.api#Unit",
        "traits": {
            "smithy.api#enumValue": "ACTIVE"
        }
    },
    "FAILED_WORKFLOW": {
        "target": "smithy.api#Unit",
        "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
        }
    },
    "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
            "smithy.api#enumValue": "DELETED"
        }
    },
    "DISABLED": {
        "target": "smithy.api#Unit",
         "traits": {
            "smithy.api#enumValue": "DISABLED"
        }
    },
    "WORKFLOW_COMPLETED": {
        "target": "smithy.api#Unit",
        "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
        }
    }
    }
},
"com.amazonaws.licensemanager#RedirectException": {
    "type": "structure",
    "members": {
        "Location": {
            "target": "com.amazonaws.licensemanager#Location",
```

```
          "traits": {
             "smithy.api#httpHeader": "Location"
          }
       },
       "Message": {
          "target": "com.amazonaws.licensemanager#Message"
       }
    },
    "traits": {
       "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
        "smithy.api#error": "client",
       "smithy.api#httpError": 308
    }
 },
 "com.amazonaws.licensemanager#RejectGrant": {
    "type": "operation",
    "input": {
       "target": "com.amazonaws.licensemanager#RejectGrantRequest"
    },
    "output": {
       "target": "com.amazonaws.licensemanager#RejectGrantResponse"
    },
    "errors": [
       {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
       },
       {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
       },
       {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
       },
       {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
       },
       {
          "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
       },
       {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
       },
       {
          "target": "com.amazonaws.licensemanager#ValidationException"
       }
    ],
    "traits": {
       "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
```

```
          }
        },
      "com.amazonaws.licensemanager#RejectGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}

}
        },
      "com.amazonaws.licensemanager#RejectGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#RenewType":
{
        "type": "enum",
        "members": {
```

```
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "WEEKLY": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Weekly"
            }
          },
          "MONTHLY": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Monthly"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#ReportContext": {
        "type": "structure",
        "members": {
          "licenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {

        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
        }
      },
      "com.amazonaws.licensemanager#ReportFrequency": {
        "type": "structure",
        "members": {
          "value": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
              "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
            }
          },
          "period": {
```

```
            "target": "com.amazonaws.licensemanager#ReportFrequencyType",

            "traits": {
               "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
            }
         }
      },
      "traits": {
         "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
      }
   },
   "com.amazonaws.licensemanager#ReportFrequencyType": {
      "type": "enum",
      "members": {
         "DAY": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "DAY"
            }
         },
         "WEEK": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "WEEK"
            }
         },
         "MONTH": {
            "target": "smithy.api#Unit",
            "traits":
{
               "smithy.api#enumValue": "MONTH"
            }
         }
      }
   },
   "com.amazonaws.licensemanager#ReportGenerator": {
      "type": "structure",
      "members": {
         "ReportGeneratorName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
               "smithy.api#documentation": "<p>Name of the report generator.</p>"
            }
         },
         "ReportType": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
```

```
          "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
        }
      },
      "ReportContext": {
        "target": "com.amazonaws.licensemanager#ReportContext",
        "traits": {
          "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
        }
      },
      "ReportFrequency": {
        "target": "com.amazonaws.licensemanager#ReportFrequency",
        "traits": {
          "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
        }
      },
      "LicenseManagerReportGeneratorArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
        }
      },
      "LastRunStatus": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
        }
      },
      "LastRunFailureReason":
{
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
        }
      },
      "LastReportGenerationTime": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
        }
      },
      "ReportCreatorAccount": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
        }
      },
```

```
        "Description": {
            "target": "com.amazonaws.licensemanager#String",

    "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
        },
        "S3Location": {
            "target": "com.amazonaws.licensemanager#S3Location",
            "traits": {
                "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
            }
        },
        "CreateTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Time the report was created.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
                "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
            }

    }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportGeneratorList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#ReportGenerator"
        }
    },
    "com.amazonaws.licensemanager#ReportGeneratorName": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "com.amazonaws.licensemanager#ReportType": {
```

```
    "type": "enum",
    "members": {
      "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
        }
      },
      "LICENSE_CONFIGURATION_USAGE_REPORT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "LicenseConfigurationUsageReport"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#ReportTypeList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ReportType"
    }
  },
  "com.amazonaws.licensemanager#ResourceInventory": {
    "type": "structure",
    "members": {
      "ResourceId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits":
{
          "smithy.api#documentation": "<p>Type of resource.</p>"
        }
      },
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "Platform": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
```

```
                          "smithy.api#documentation": "<p>Platform of the resource.</p>"
                      }
                  },
                  "PlatformVersion": {
                      "target": "com.amazonaws.licensemanager#String",
                      "traits": {
                          "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
                      }
                  },
                  "ResourceOwningAccountId":
{
                      "target": "com.amazonaws.licensemanager#String",
                      "traits": {
                          "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
                      }
                  }
              },
              "traits": {
                  "smithy.api#documentation": "<p>Details about a resource.</p>"
              }
          },
          "com.amazonaws.licensemanager#ResourceInventoryList": {
              "type": "list",
              "member": {
                  "target": "com.amazonaws.licensemanager#ResourceInventory"
              }
          },
          "com.amazonaws.licensemanager#ResourceLimitExceededException": {
              "type": "structure",
              "members": {
                  "Message": {
                      "target": "com.amazonaws.licensemanager#Message"
                  }
              },
              "traits": {
                  "aws.protocols#awsQueryError": {
                      "code": "ResourceLimitExceeded",

                      "httpResponseCode": 400
                  },
                  "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
                  "smithy.api#error": "client",
                  "smithy.api#httpError": 400
              }
          },
          "com.amazonaws.licensemanager#ResourceNotFoundException": {
              "type": "structure",
              "members": {
```

```json
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InvalidResource.NotFound",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>The resource cannot be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#ResourceType": {
      "type": "enum",
      "members": {
        "EC2_INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_INSTANCE"
          }
        },
        "EC2_HOST": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_HOST"
          }
        },
        "EC2_AMI": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_AMI"
          }
        },
        "RDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "RDS"
          }
        },
        "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
          }
        }
      }
```

```
        },
        "com.amazonaws.licensemanager#S3Location": {
            "type": "structure",
            "members": {
                "bucket": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
                    }
                },
                "keyPrefix": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
            }
        },
        "com.amazonaws.licensemanager#ServerInternalException": {
            "type": "structure",
            "members": {
                "Message": {
                    "target": "com.amazonaws.licensemanager#Message"
                }
            },
            "traits": {
                "aws.protocols#awsQueryError": {
                    "code": "InternalError",
                    "httpResponseCode": 500
                },
                "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
                "smithy.api#error": "server",
                "smithy.api#httpError": 500
            }
        },
        "com.amazonaws.licensemanager#SignedToken": {
            "type": "string",
            "traits": {
                "smithy.api#length": {
                    "min": 4096
                }
            }
        },
        "com.amazonaws.licensemanager#StatusReasonMessage": {
```

```
            "type": "string",
            "traits":
{
                "smithy.api#length": {
                    "min": 0,
                    "max": 400
                },
                "smithy.api#pattern": "^[\\s\\S]+$"
            }
        },
        "com.amazonaws.licensemanager#String": {
            "type": "string"
        },
        "com.amazonaws.licensemanager#StringList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#String"
            }
        },
        "com.amazonaws.licensemanager#Tag": {
            "type": "structure",
            "members": {
                "Key": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Tag key.</p>"
                    }
                },
                "Value": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation":
"<p>Tag value.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#TagKeyList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#String"
            }
        },
        "com.amazonaws.licensemanager#TagList": {
            "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanager#Tag"
        }
    },
    "com.amazonaws.licensemanager#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#TagResourceResponse"
      },
      "errors": [
        {
          "target":
"com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#TagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits":
{
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
```

```
          "traits": {
            "smithy.api#documentation": "<p>One or more tags.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
},
"com.amazonaws.licensemanager#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#TokenData": {
  "type": "structure",
  "members": {
    "TokenId": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token ID.</p>"
      }
    },
    "TokenType": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
      }
    },
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
      }
    },
    "ExpirationTime": {
      "target": "com.amazonaws.licensemanager#ISO8601DateTime",
      "traits": {
        "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
format.</p>"
      }
    },
    "TokenProperties": {
      "target": "com.amazonaws.licensemanager#MaxSize3StringList",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Data specified by the caller.</p>"
          }
        },
        "RoleArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
          }
        }
      },
      "traits": {

        "smithy.api#documentation": "<p>Describes a token.</p>"
      }
    },
    "com.amazonaws.licensemanager#TokenList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#TokenData"
      }
    },
    "com.amazonaws.licensemanager#TokenString": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 4096
        },
        "smithy.api#pattern": "^\\S+$"
      }
    },
    "com.amazonaws.licensemanager#TokenType": {
      "type": "enum",
      "members": {
        "REFRESH_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REFRESH_TOKEN"
          }
```

```
          }
        }
      },
      "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
 {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
           },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
        }
      },
```

```
        "com.amazonaws.licensemanager#UntagResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanager#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
          },
          "output": {

"target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
    "type": "structure",
    "members": {
      "LicenseConfigurationArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseConfigurationStatus": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
        "traits": {
          "smithy.api#documentation": "<p>New status of the license configuration.</p>"
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {

  "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
  configuration is licenseAffinityToHost.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
        }
      },
      "LicenseCountHardLimit": {
```

```
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>New description of the license configuration.</p>"
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>New product information.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        }
      },
      "traits":
{
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportGeneratorName": {
```

```
            "target": "com.amazonaws.licensemanager#ReportGeneratorName",
            "traits": {
              "smithy.api#documentation": "<p>Name of the report generator.</p>",
              "smithy.api#required": {}
            }
          },
          "Type": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
              "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n        <ul>\n          <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n          </li>\n          <li>\n          <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n          </li>\n        </ul>",
              "smithy.api#required": {}
            }
          },
          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>The report context.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits":
{
              "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientRequestToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#input": {}
          }
      },
      "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
  {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
      },
      "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },

            {
              "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
            },
            {
              "target": "com.amazonaws.licensemanager#LicenseUsageException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
as they send license configurations to the operation that creates the resource.</p>"
```

```
            }
        },

        "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
            "type": "structure",
            "members": {
                "ResourceArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
                        "smithy.api#required": {}
                    }
                },
                "AddLicenseSpecifications": {
                    "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                    "traits": {
                        "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
                    }
                },
                "RemoveLicenseSpecifications": {
                    "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                    "traits": {
                        "smithy.api#documentation":
"<p>ARNs of the license configurations to remove.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateServiceSettings": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
            },
            "errors": [
```

```
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "S3BucketArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where the License
 Manager information is stored.</p>"
            }
          },
          "SnsTopicArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
            }
          },
          "OrganizationConfiguration": {
            "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
            }
          },
          "EnableCrossAccountsDiscovery": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
    "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
    "com.amazonaws.licensemanager#UsageOperation": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          }
        }
      },
    "com.amazonaws.licensemanager#ValidationException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

      "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      }
    }
  }
}
```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"

```go
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version
 of the license.
 //
 // This member is required.
 SourceVersion *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus
```

```go
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack,
options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
```

```go
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "DeleteLicense",
	}
}
```

1.21 collector-pdata 1.0.0

1.21.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the

editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding
those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions

   for use, reproduction, or distribution of Your modifications, or
    for any such Derivative Works as a whole, provided Your use,
    reproduction, and distribution of the Work otherwise complies with
    the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
    any Contribution intentionally submitted for inclusion in the Work
    by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.22 golang 1.21.5

1.22.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit
(http://www.openssl.org/)
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).
This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1.22.2 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct,
indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "{}"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.
The Go source code and supporting files in this directory
are covered by the usual Go license (see ../../../../LICENSE).

When building with GOEXPERIMENT=boringcrypto, the following applies.

The goboringcrypto_linux_amd64.syso object file is built
from BoringSSL source code by build/build.sh and is covered
by the BoringSSL license reproduced below and also at
https://boringssl.googlesource.com/boringssl/+/fips-20190808/LICENSE.

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL
licensing. Files that are completely new have a Google copyright and an ISC
license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium:
https://cla.developers.google.com/clas

Some files from Intel are under yet another license, which is also included
underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the
OpenSSL License and the original SSLeay license apply to the toolkit.
 See below
for the actual license texts. Actually both licenses are BSD-style Open Source
licenses. In case of any license issues related to OpenSSL please contact
openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from
some authors is recorded for use of their work. (This is purely for our own
record keeping.)
 27287199
 27287880
 27287883

 OpenSSL License

/* ==
* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of
 the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*   Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are
 not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must include an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE

notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the
distribution.
#
* Neither the name of the Intel Corporation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.
#
#
THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION
 ""AS IS"" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache License 2.0
Copyright 2010-2021 Mike Bostock

Permission to use, copy, modify, and/or distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright notice
and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF
THIS SOFTWARE.
People who have agreed to one of the CLAs and can contribute patches.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.
#
https://developers.google.com/open-source/cla/individual
https://developers.google.com/open-source/cla/corporate
#
Names should be added to this file as:
Name <email address>
Raul Silvera <rsilvera@google.com>
Tipp Moseley <tipp@google.com>

Hyoun Kyu Cho <netforce@google.com>

Martin Spier <spiermar@gmail.com>

Taco de Wolff <tacodewolff@gmail.com>

Andrew Hunter <andrewhhunter@gmail.com>

other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct,
indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
    incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.
Copyright (c) 2015 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2009-2017 Andrea Leofreddi <a.leofreddi@vleo.net>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

  1. Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.
  2. Redistributions in binary form must reproduce the above copyright
     notice, this list of conditions and the following disclaimer in the
     documentation and/or other materials provided with the distribution.
  3. Neither the name of the copyright holder nor the names of its
     contributors may be used to endorse or promote products derived from
     this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY
EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY

1.23 aws-aws-sdk-go-v2 1.10.3

1.23.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
```

```go
	return nil,
err
}

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

// The type of license to associate with the workspace.
//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
	return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err !=
nil {
	return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
```

```
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack);
err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
```

```go
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack,
	options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:       region,
		ServiceID:    ServiceID,
		OperationName: "AssociateLicense",
	}
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

```yaml
name: License Scan

on: [pull_request]

jobs:
 licensescan:
  name: License Scan
  runs-on: ubuntu-latest
  strategy:
   matrix:
    python-version: [3.9]

  steps:
```

```yaml
    - name: Checkout target
      uses: actions/checkout@v2
      with:
        path: sdkbase
        ref: ${{ github.base_ref }}
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
        path: new-ref
        fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }} ${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
```

```go
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
if params == nil {
params = &CheckInLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
if err != nil {
return nil, err
}

out := result.(*CheckInLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckInLicenseInput struct {

// License consumption token.
//
// This member is required.
LicenseConsumptionToken *string

// License beneficiary.
Beneficiary
*string

noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
```

```go
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```go
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
		return
err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckInLicense",
	}
}
```

```json
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
            "type": "structure",
```

```
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
          }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
        "type": "structure",
        "members": {}
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n      Organizations integration has been enabled, and
 which Regions data will be aggregated from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
```

```
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays
the aggregated data for Linux\n      subscriptions.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
            "AmiId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
                }
            },
            "InstanceID": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
                }
            },
            "InstanceType": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The instance type of the resource.</p>"
                }

            },
            "AccountID": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the instance.</p>"
                }
            },
            "Region": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
```

```
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n       operation values</a> in the <i>License Manager User
Guide</i>.</p>"
            }
          },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
              "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n       operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
          },
          "LastUpdatedTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
```

```
              "target": "smithy.api#String"
          }
      },
      "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
      }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
          {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
          },
          {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
          },
          {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
          },
          {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
      ],

      "traits": {
          "aws.api#service": {
              "sdkId": "License Manager Linux Subscriptions",
              "arnNamespace": "license-manager-linux-subscriptions",
              "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
              "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
              "additionalAllowedHeaders": [
                  "*",
                  "content-type",
                  "x-amz-content-sha256",
                  "x-amz-user-agent",
                  "x-amzn-platform-id",
                  "x-amzn-trace-id",
                  "content-length",
                  "x-api-key",
                  "authorization",
                  "x-amz-date",
```

```
        "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
            ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n     Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
                "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
                    "documentation": "The
 AWS region used to dispatch the request.",
                    "type": "String"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

                    "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }
```

```
                  },
             "rules": [
                {
                   "conditions": [
                      {
                         "fn": "isSet",
                         "argv": [
                            {
                               "ref": "Endpoint"
                            }
                         ]
                      }
                   ],
                   "type": "tree",
                   "rules": [
                      {

                      "conditions": [
                            {
                               "fn": "booleanEquals",
                               "argv": [
                                  {
                                     "ref": "UseFIPS"
                                  },
                                  true
                               ]
                            }
                         ],
                         "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                         "type": "error"
                      },
                      {
                         "conditions": [],
                         "type": "tree",
                         "rules": [
                            {

                         "conditions": [
                               {
                                  "fn": "booleanEquals",
                                  "argv": [
                                     {
                                        "ref": "UseDualStack"
                                     },
                                     true
                                  ]
                               }
                            ],
```

```
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                        "type": "error"
                    },
                    {
                        "conditions": [],

            "endpoint": {
                            "url": {
                                "ref": "Endpoint"
                            },
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    },
    {
        "conditions": [],
        "type": "tree",
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",

"argv": [
                            {
                                "ref": "Region"
                            }
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "aws.partition",
                                "argv": [
                                    {
                                        "ref": "Region"
                                    }
                                ],
```

```
            "assign": "PartitionResult"
                                    }
                          ],
                          "type": "tree",
                          "rules": [
                            {
                              "conditions": [
                                {
                                  "fn": "booleanEquals",
                                  "argv": [
                                    {
                                      "ref": "UseFIPS"
                                    },
                                    true
                                  ]
                                },

                  {
                                  "fn": "booleanEquals",
                                  "argv": [
                                    {
                                      "ref": "UseDualStack"
                                    },
                                    true
                                  ]
                                }
                              ],
                              "type": "tree",
                              "rules": [
                                {
                                  "conditions": [
                                    {

                              "fn": "booleanEquals",
                                      "argv": [
                                        true,
                                        {
                                          "fn": "getAttr",
                                          "argv": [
                                            {
                                              "ref": "PartitionResult"
                                            },
                                            "supportsFIPS"
                                          ]

      }
                                      ]
                                    },
```

```
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  true,
                                  {
                                    "fn": "getAttr",
                                    "argv": [
                                      {
                                        "ref": "PartitionResult"
                                      },

                                "supportsDualStack"
                                    ]
                                  }
                                ]
                              }
                            ],
                            "type": "tree",
                            "rules": [
                              {
                                "conditions": [],
                                "type": "tree",
                                "rules": [
                                  {

                      "conditions": [],
                                    "endpoint": {
                                      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                      "properties": {},
                                      "headers": {}
                                    },
                                    "type": "endpoint"
                                  }
                                ]
                              }
                            ]

                },
                          {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                            "type": "error"
                          }
                        ]
                    },
                    {
```

```
"conditions": [
    {
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    }
],
"type": "tree",
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            }
        ],
        "type": "tree",
        "rules": [
            {
                "conditions": [],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
```

```
                },
                  "type": "endpoint"
                }
              ]
            }
          ]
        },
        {
          "conditions": [],
          "error": "FIPS is enabled but this partition does not support FIPS",
          "type": "error"
        }
      ]

    },
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }
                      ],
                      "type": "tree",
                      "rules": [
                        {

                     "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
                                      "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                  ]
                                }
                              ]
                            }
                          ],
```

```
"type": "tree",
"rules": [
  {
    "conditions": [],
    "type": "tree",
    "rules": [
      {
        "conditions": [],

        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ]
  }
],

{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",
  "type": "error"
}
]
},
{
  "conditions": [],
  "type": "tree",
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ]
}
]
}
]
```

```
            },
            {
              "conditions": [],
              "error": "Invalid Configuration: Missing Region",
              "type": "error"
            }
          ]
        }
      ]
    },
    "smithy.rules#endpointTests": {
      "testCases": [
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
```

```
        "UseFIPS": false,
         "UseDualStack": true
      }
   },
   {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
         "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
         }
      },
      "params": {
         "Region": "us-east-1",
         "UseFIPS": false,
         "UseDualStack": false
      }
   },
   {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
         "endpoint": {

"url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
         }
      },
      "params": {
         "Region": "cn-north-1",
         "UseFIPS": true,
         "UseDualStack": true
      }
   },
   {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
         "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
         }
      },
      "params": {
         "Region": "cn-north-1",
         "UseFIPS": true,
          "UseDualStack": false
      }
   },
   {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
         "endpoint": {
```

                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint":

{

                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": true,
               "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }

```
            },
            {
              "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "endpoint":
{
                    "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
                  }
              },
              "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
                  }
              },
              "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                  }
              },
              "params": {
                "Region": "us-iso-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {

  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {

           "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
```

```
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
              "expect": {
                "error": "DualStack is enabled but this partition does not support DualStack"
              },
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                }
              },
              "params": {
                "Region": "us-isob-east-1",

      "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
```

disabled",

          "expect": {
                "endpoint": {
                   "url": "https://example.com"
                }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,

      "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "Missing region",
          "expect": {
            "error": "Invalid Configuration: Missing Region"
          }

    }
      ],
      "version": "1.0"
     }

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
        "type": "structure",
        "members": {
          "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
              "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type":
"operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
          "smithy.api#http": {

    "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by
 the name of\n        <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n        <p>The valid names for this filter
are:</p>\n        <ul>\n          <li>\n            <p>\n                <code>AmiID</code>\n            </p>\n
</li>\n          <li>\n            <p>\n                <code>InstanceID</code>\n            </p>\n          </li>\n
<li>\n            <p>\n                <code>AccountID</code>\n            </p>\n          </li>\n          <li>\n
<p>\n                <code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
```

<code>UsageOperation</code>\n   </p>\n   \n   \n   <p>\n   <code>ProductCode</code>\n   </p>\n   \n   \n   <p>\n   <code>InstanceType</code>\n   </p>\n   \n   \n   <p>The valid Operators for this filter are:</p>\n   \n   \n   <p>\n   <code>contains</code>\n   </p>\n   \n   \n   <p>\n   <code>equals</code>\n   </p>\n   \n   \n   <p>\n   <code>Notequal</code>\n   </p>\n   \n   "
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                        "smithy.api#length": {
                            "min": 1,
                            "max": 16384
                        }
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
            "type": "structure",
            "members": {
                "Instances": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                    "traits": {
                        "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            }

```
        },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n      organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }

        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't
```

match a certain subscription's name.</p>\n   <p>The valid names for this filter are:</p>\n   \n   \n   <p>\n   <code>Subscription</code>\n   </p>\n   \n   \n <p>The valid Operators for this filter are:</p>\n   \n   \n   <p>\n   <code>contains</code>\n   </p>\n   \n   \n   <p>\n   <code>equals</code>\n   </p>\n   <p>\n   <code>Notequal</code>\n   </p>\n   "
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                        "smithy.api#length": {

                            "min": 1,
                            "max": 16384
                        }
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
            "type": "structure",
            "members": {
                "Subscriptions": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                    "traits": {
                        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            }

```
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "EQUAL",
                        "value": "Equal",
                        "documentation": "Equal operator"
                    },
                    {
                        "name": "NOT_EQUAL",
                        "value": "NotEqual",
                        "documentation": "Not equal operator"
                    },
                    {
                        "name": "CONTAINS",
                        "value": "Contains",
                        "documentation": "Contains operator"
                    }
                ],
                "smithy.api#length": {
                    "min": 1,
                    "max": 20
                }
            }
        },

        "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "Enabled",
                        "value": "Enabled",
                        "documentation": "Enabled OrganizationIntegration"
                    },
                    {
                        "name": "Disabled",
                        "value": "Disabled",
                        "documentation": "Disabled OrganizationIntegration"
                    }
                ]
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
            "type": "list",
            "member": {
```

```
              "target": "smithy.api#String"
          }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [

                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
                    "value": "Successful",
                    "documentation": "Successful status"
                },
                {
                    "name": "Failed",
                    "value": "Failed",
                    "documentation": "Failed status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits":
{
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
```

```
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"

        }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
```

```json
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
                "smithy.api#error": "client"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
            },

            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/subscription/UpdateServiceSettings"
                },
                "smithy.api#idempotent": {}
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
            "type": "structure",
            "members": {
                "LinuxSubscriptionsDiscovery": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required": {}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>",

  "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n     allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },

        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
```

applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation":
"<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n   subscriptions.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.

```go
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
	if params == nil {
		params = &ListLicensesInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*ListLicensesOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type ListLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//   - Beneficiary
	//   - ProductSKU
	//   - Fingerprint
	//
	//   - Status
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListLicensesOutput struct {

	// License details.
	Licenses []types.License

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
```

```
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	if
	err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
```

```go
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
	err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:
			ServiceID,
		OperationName: "ListLicenses",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana
```

```go
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 //
 This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
```

```go
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "DisassociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
```

```go
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
{
	return &awsmiddleware.RegisterServiceMetadata{
```

```go
  Region:        region,
  ServiceID:      ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type
 CheckoutLicenseInput struct {

 // Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
```

```go
	// the request.
	//
	// This member is required.
	ClientToken *string

	// License entitlements.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Key fingerprint identifying the license.
	//
	// This member is required.
	KeyFingerprint *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// License beneficiary.
	Beneficiary *string

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

	// Checkout type.
	CheckoutType types.CheckoutType

	// Allowed license entitlements.
	EntitlementsAllowed []types.EntitlementData

	// Date and time at which the license checkout expires.
	Expiration *string

	// Date and time at which the license checkout is issued.
	IssuedAt *string

	// Amazon Resource
	Name (ARN) of the checkout license.
	LicenseArn *string

	// License consumption token.
```

```
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil
 {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
```

```
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
     return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
     return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
     return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack);
    err != nil {
     return err
    }
    if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
     return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
    middleware.Before); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
    }
```

```go
func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "CheckoutLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier
 that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string
```

```go
	// Digital signature method. The possible value is JSON Web Signature (JWS)
	// algorithm PS384. For more information, see RFC 7518 Digital Signature with
	// RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
	//
	// This member is required.
	DigitalSignatureMethod types.DigitalSignatureMethod

	// License entitlements. Partial checkouts are not supported.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Amazon Resource Name (ARN) of the license. The license must use the borrow
	// consumption configuration.
	//
	// This member is required.
	LicenseArn *string

	// Information about constraints.
	CheckoutMetadata []types.Metadata

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

	// Information about constraints.
	CheckoutMetadata []types.Metadata

	// Allowed license entitlements.
	EntitlementsAllowed
	[]types.EntitlementData

	// Date and time at which the license checkout expires.
	Expiration *string

	// Date and time at which the license checkout is issued.
	IssuedAt *string

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License consumption token.
	LicenseConsumptionToken *string
```

```go
	// Node ID.
	NodeId *string

	// Signed token.
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
	if err != nil {
		return
err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
```

```
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return
    err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
    middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
    }
```

```go
func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide
```

to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity

```go
    *types.DatetimeRange

    // Information about the license.
    LicenseMetadata []types.Metadata

    noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add
    protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
```

```go
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
}
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
```

```go
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName:
 "CreateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
```

```go
	return out, nil
}

type GetLicenseInput struct {

	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// License version.
	Version
	*string

	noSmithyDocumentSerde
}

type GetLicenseOutput struct {

	// License details.
	License *types.License

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return
err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addOpGetLicenseValidationMiddleware(stack);
err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
```

```
    return err
   }
  if err = addResponseErrorMiddleware(stack); err != nil {
   return err
   }
  if err = addRequestResponseLogging(stack, options); err != nil {
   return err
   }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
   return err
   }
  return nil
 }

 func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
  return &awsmiddleware.RegisterServiceMetadata{
   Region:       region,
   ServiceID:     ServiceID,
   OperationName: "GetLicense",
   }
 }
```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //  - ProductSKU
 //  - Status
 //  - Fingerprint
 //  - IssuerName
 //  - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
```

```go
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

	// Received license details.
	Licenses []types.GrantedLicense

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
```

```
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
    return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack);
    err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
    middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
```

```
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership
of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

 4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
 incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
```

```
        "type": "structure",
        "members": {
          "DirectoryId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related
```

costs will take 48\n   hours to appear for billing periods that haven't closed (marked as Pending billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
      monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p>\n   </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {

```
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
           "traits": {
             "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
             "smithy.api#required": {}
           }
        }
     }
   },
   "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
     "type": "integer"
   },
   "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
     "type": "structure",
     "members": {
       "message": {
         "target": "smithy.api#String"
       }
     },
     "traits": {
       "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n      resource.</p>",
       "smithy.api#error": "server"
     }
   },
   "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
     "type": "operation",
     "input": {
       "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
     },
     "output": {
       "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
     },
     "errors": [
       {
         "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
       },
       {
         "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
       },
       {
         "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
       },
       {
         "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
       },
       {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/DeregisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
```

```
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/user/DisassociateUser"
                },
                "smithy.api#idempotent": {}
            }

        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
```

```
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },

    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
```

```
        "members": {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used
 to return more specific results from a describe\n      operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n      attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
```

```
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n    settings such as the subnets to provision VPC endpoints.</p>",
                    "smithy.api#required":
{}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "FailureMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
```

```
            }
          },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status
 of an EC2 instance resource.</p>",
                "smithy.api#required": {}
              }
            },
            "Products": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
                "smithy.api#required": {}
              }
            },
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
              }

          }
            },
          "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
          "type": "list",
```

```
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
           "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
```

```json
        },
        "AssociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
          }
        },
        "DisassociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",

      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
```

      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
      }

    },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
      },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
          "x-amzn-trace-id",

```
                "content-length",
                "x-api-key",
                "authorization",
                "x-amz-date",
                "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
              ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"

]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack
 endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
```

```
              "type": "String"
            }

        },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

                        true
                      ]
                    }
                  ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
```

```
          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                          "ref": "Endpoint"
                      },
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                }
              ]
          },
          {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [

    {
                          "ref": "Region"
                      }
                  ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                      {
                          "fn": "aws.partition",
                          "argv": [
                            {
                                "ref": "Region"
                            }
                          ],
                          "assign": "PartitionResult"
                      }
                    ],
                    "type": "tree",

"rules": [
                        {
                            "conditions": [
                                {
```

```
                              "fn": "booleanEquals",
                              "argv": [
                                {
                                   "ref": "UseFIPS"
                                },
                                true
                              ]
                          },
                          {
                              "fn": "booleanEquals",
                              "argv": [
                                {
                                   "ref": "UseDualStack"


                                },
                                true
                              ]
                          }
                      ],
                      "type": "tree",
                      "rules": [
                          {
                              "conditions": [
                                {
                                   "fn": "booleanEquals",
                                   "argv": [
                                      true,
                                      {
                                         "fn": "getAttr",

    "argv": [

                                            {
                                               "ref": "PartitionResult"
                                            },
                                            "supportsFIPS"
                                         ]
                                      }
                                   ]
                                },
                                {
                                   "fn": "booleanEquals",
                                   "argv": [
                                      true,
                                      {

          "fn": "getAttr",
                                         "argv": [
                                            {
```

```
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]
                        }
                      ]
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {

        "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                              },
                              "type": "endpoint"
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                     "type": "error"
                  }
                ]
              },
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "type": "tree",

      "rules": [
                  {
                    "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                      ]
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [],
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }
              ]
            },
            {
              "conditions": [],
              "error": "FIPS is enabled but this partition does not support FIPS",
              "type": "error"
            }
          ]
        },
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
```

```
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {

                            "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                              ]
                          }
                        ]
                      }
                    ],
                    "type": "tree",
                    "rules": [
                      {

                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "error": "DualStack is enabled but this partition does
not support DualStack",
                    "type": "error"
                  }
                ]
              },
              {
```

```
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ]
                }
            ]
        },
        {
            "conditions":
[],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
```

```
                        }
                    },
                    {
                        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                            }
                        },

                "params": {
                        "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                        }
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                        }
                    },
                    "params": {
                        "Region": "cn-north-1",
                        "UseFIPS": true,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
```

```
                        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                    }
                },

        "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },

        "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
```

```
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                        }
                    },
                    "params": {
                        "Region": "us-gov-east-1",
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                        }
                    },
                    "params": {
                        "Region": "us-gov-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
                        }
                    },

        "params": {
                        "Region": "us-gov-east-1",
                        "UseFIPS": false,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
```

```
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect":
{
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack":
true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
```

```
                                "expect": {
                                  "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
                                },
                                "params": {
                                  "Region": "us-isob-east-1",
                                  "UseFIPS": true,
                                  "UseDualStack": true
                                }
                              },
                              {
                                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                                "expect": {
                                  "endpoint": {
                                    "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                                  }
                                },
                                "params": {
                                  "Region": "us-isob-east-1",
                                  "UseFIPS": true,
                                  "UseDualStack": false
                                }
                              }

                    },
                              {
                                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
                                "expect": {
                                  "error": "DualStack is enabled but this partition does not support DualStack"
                                },
                                "params": {
                                  "Region": "us-isob-east-1",
                                  "UseFIPS": false,
                                  "UseDualStack": true
                                }
                              },
                              {
                                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                                "expect": {
                                  "endpoint": {
                                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                                  }
                                },
                                 "params": {
                                  "Region": "us-isob-east-1",
                                  "UseFIPS": false,
                                  "UseDualStack": false
                                }
                              },
```

```json
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
```

```
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
           },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
```

```
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated":
{
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
```

```
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
  "type":
"structure",
  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
```

              "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
              "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
              },
              "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
            "type": "structure",
            "members": {
              "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name

```
of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {

"smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n    more sets of key-value pairs that you specify.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
                }
            },

        "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
```

```
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
```

```
            "items": "InstanceUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
```

```
        "InstanceUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {

      "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }

    },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for a user.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusMessage": {
```

```
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
          }
        },
        "SubscriptionEndDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
```

```json
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/RegisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n    subnets to provision VPC endpoints.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
        "type": "string",

    "traits": {
        "smithy.api#length": {
          "min": 5,
          "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
      }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
        "type": "structure",
        "members": {
```

```
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n     your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n     subnets to provision VPC endpoints, and the security group ID that
 is associated with the VPC\n     endpoints. The security group should permit inbound TCP port 1688
communication from resources\n     in the VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
```

```
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
       },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n        <note>\n          <p>Your estimated bill for charges on the number of users and related costs
will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
 For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n        monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StartProductSubscription"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
           "traits": {
             "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
             "smithy.api#required": {}
           }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
       "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

    },
        {
```

```
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required":
{}
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the user.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation":
"<p>Metadata that describes the start product subscription operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
```

```
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity\n      provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
```

```
                    "traits": {
                        "smithy.api#required": {}
                    }
                },
                "Product": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                        "smithy.api#required": {}
                    }
                },
                "UpdateSettings": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                    "traits": {
                        "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n     update any combination of settings in a single operation such as the:</p>\n
<ul>\n          <li>\n               <p>Subnets which you want to add to provision VPC endpoints.</p>\n          </li>\n
<li>\n               <p>Subnets which you want to remove the VPC endpoints from.</p>\n          </li>\n
<li>\n               <p>Security group ID which permits traffic to the VPC endpoints.</p>\n          </li>\n        </ul>",
                        "smithy.api#required": {}
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
            "type": "structure",
            "members": {
                "IdentityProviderSummary": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                    "traits": {
                        "smithy.api#required": {}
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
            "type": "structure",
            "members": {
                "AddSubnets": {

                    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                    "traits": {
                        "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n     require connectivity to activation servers.</p>",
                        "smithy.api#required": {}
                    }
                },
                "RemoveSubnets": {
```

```
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
              }
            },
            "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                "smithy.api#documentation": "<p>A security group ID
  that allows inbound TCP port 1688 communication between resources in\n    your VPC and the VPC endpoints for
  activation servers.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
  settings such as\n    the subnets to provision VPC endpoints.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
          }
        }
      }
    }
    {
      "smithy": "2.0",
      "metadata": {
        "suppressions": [
          {
            "id": "HttpMethodSemantics",
            "namespace": "*"
          },
          {
            "id": "HttpResponseCodeSemantics",
            "namespace": "*"
          },
          {
            "id": "PaginatedTrait",
```

```
          "namespace": "*"
        },
        {
          "id": "HttpHeaderTrait",
          "namespace": "*"
        },
        {
          "id": "HttpUriConflict",
          "namespace": "*"
        },
        {
          "id": "Service",
          "namespace": "*"
        }
      ]
    },
    "shapes": {
      "com.amazonaws.licensemanager#AWSLicenseManager": {
        "type": "service",
        "version": "2018-08-01",
        "operations": [
          {
            "target": "com.amazonaws.licensemanager#AcceptGrant"
          },

  {
            "target": "com.amazonaws.licensemanager#CheckInLicense"
          },
          {
            "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
          },
          {
            "target": "com.amazonaws.licensemanager#CheckoutLicense"
          },
          {
            "target": "com.amazonaws.licensemanager#CreateGrant"
          },
          {
            "target": "com.amazonaws.licensemanager#CreateGrantVersion"
          },
          {
            "target": "com.amazonaws.licensemanager#CreateLicense"
          },
          {
            "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
          },
          {
            "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
```

```
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateToken"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
      "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    },
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
    {
      "target": "com.amazonaws.licensemanager#GetGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
    },
    {
```

```
      "target": "com.amazonaws.licensemanager#GetLicenseUsage"
    },
    {
      "target": "com.amazonaws.licensemanager#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#ListDistributedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
    },
    {

      "target": "com.amazonaws.licensemanager#ListLicenseVersions"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListResourceInventory"
    },
```

```
    {
      "target": "com.amazonaws.licensemanager#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTokens"
    },
    {
      "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#RejectGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager",
      "arnNamespace": "license-manager",
      "cloudFormationName": "LicenseManager",
      "cloudTrailEventSource": "licensemanager.amazonaws.com",
      "endpointPrefix": "license-manager"
    },
    "aws.auth#sigv4": {
      "name": "license-manager"
    },
    "aws.protocols#awsJson1_1": {},
    "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n        Amazon Web Services accounts and on-premises servers.</p>",
    "smithy.api#title": "AWS License Manager",
    "smithy.api#xmlNamespace": {
      "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
```

```
                },
            "smithy.rules#endpointRuleSet": {
                "version": "1.0",
                "parameters": {
                    "Region": {
                        "builtIn": "AWS::Region",
                        "required": false,
                        "documentation": "The AWS region used to dispatch the request.",
                        "type": "String"
                    },
                    "UseDualStack": {
                        "builtIn": "AWS::UseDualStack",
                        "required": true,
                        "default": false,
                        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                        "type": "Boolean"
                    },
                    "UseFIPS": {
                        "builtIn": "AWS::UseFIPS",
                        "required": true,
                        "default": false,
                        "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
                        "type": "Boolean"
                    },
                    "Endpoint": {
                        "builtIn": "SDK::Endpoint",
                        "required": false,
                        "documentation": "Override the endpoint used to send this request",
                        "type": "String"
                    }
                },
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "isSet",
                                "argv": [
                                    {
                                        "ref": "Endpoint"
                                    }
                                ]

                            }
                        ],
                        "type": "tree",
```

```
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {

"conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
]
},
{
    "conditions": [
        {
```

```json
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "type": "tree",
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],
              "type": "tree",
              "rules": [
```

```
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsFIPS"
                                        ]
                                    }
                                ]
                            },
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
```

```
                              ]
                        },
                        {
                          "conditions": [],
                          "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                          "type": "error"
                        }
                      ]
                  },
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",

                   "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      }
                    ],
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              true,

          {
                                "fn": "getAttr",
                                "argv": [
                                  {
                                    "ref": "PartitionResult"
                                  },
                                  "supportsFIPS"
                                ]
                              }
                            ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
```

```
                            {
                                 "conditions": [],
                                 "endpoint": {
                                    "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                 },
                                 "type": "endpoint"
                            }
                        ]
                   },
                   {
                       "conditions": [],
                       "error": "FIPS is enabled but this
partition does not support FIPS",
                       "type": "error"
                   }
                ]
           },
           {
               "conditions": [
                   {
                       "fn": "booleanEquals",
                       "argv": [
                           {
                               "ref": "UseDualStack"
                           },
                           true
                       ]
                   }
               ],
              "type": "tree",
               "rules": [
                   {
                       "conditions": [
                           {
                               "fn": "booleanEquals",
                               "argv": [
                                   true,
                                   {
                                       "fn": "getAttr",
                                       "argv": [
                                           {
                                               "ref": "PartitionResult"
                                           },

    "supportsDualStack"
                                       ]
```

```
                        }
                      ]
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}

                      },
                      "type": "endpoint"
                    }
                  ]
                },
                {
                  "conditions": [],
                  "error": "DualStack is enabled but this partition does not support DualStack",
                  "type": "error"
                }
              ]
            },
            {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ]
        }
      ]
    },
    {
      "conditions": [],
      "error": "Invalid Configuration: Missing Region",
      "type": "error"
    }
  ]
},
"smithy.rules#endpointTests": {
  "testCases": [
```

```
      {
        "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint":
{
            "url": "https://license-manager.af-south-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "af-south-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-northeast-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-2.amazonaws.com"
          }
        },
        "params": {
```

```
            "Region": "ap-northeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
      },
      {
          "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager.ap-northeast-3.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }

      },
      {
          "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager.ap-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
      },
      {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager.ap-southeast-1.amazonaws.com"

}
          },
          "params": {
            "Region": "ap-southeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
      },
      {
          "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
```

```
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-southeast-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-southeast-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {

"documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-southeast-3.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-southeast-3",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ca-central-1.amazonaws.com"
          }
        },

        "params": {
          "Region": "ca-central-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.eu-central-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "eu-central-1",
```

```
              "UseFIPS": false,
              "UseDualStack": false
            }
        },
        {
            "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                 "url": "https://license-manager.eu-north-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "eu-north-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
        },
        {
            "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                 "url": "https://license-manager.eu-south-1.amazonaws.com"
              }
            },
            "params": {
              "Region":
"eu-south-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
        },
        {
            "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                 "url": "https://license-manager.eu-west-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "eu-west-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
        },
        {
            "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
            "expect": {
```

```json
          "endpoint": {
            "url": "https://license-manager.eu-west-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "eu-west-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.eu-west-3.amazonaws.com"
          }
        },
        "params": {
          "Region": "eu-west-3",
          "UseFIPS": false,

"UseDualStack": false
        }
      },
      {
        "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.me-south-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "me-south-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.sa-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "sa-east-1",
          "UseFIPS": false,
```

```
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false
              }

    },
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.us-east-2.amazonaws.com"
                }

     },
              "params": {
                "Region": "us-east-2",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
```

```
            "url": "https://license-manager-fips.us-east-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-2",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region
us-west-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-west-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-west-1.amazonaws.com"
          }
        },
        "params": {

          "Region": "us-west-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-west-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-west-2",
          "UseFIPS": false,
          "UseDualStack": false
```

```
            }
        },
        {
            "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-fips.us-west-2.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-west-2",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,

                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url":
```

```
                    "https://license-manager.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-northwest-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }

    }
            },
            {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
```

```
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },

    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-west-1.amazonaws.com"
          }
        },
```

```json
      "params": {
        "Region": "us-gov-west-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-west-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
```

```
            "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "error": "DualStack is enabled
but this partition does not support DualStack"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
```

```
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },

    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
```

```
                "UseDualStack": true
              }
            },
            {
              "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

      "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
  disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,

      "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
```

                         "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanager#AcceptGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

},
        "output": {
        "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"

```
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"

        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#AcceptGrantRequest": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#AcceptGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant
ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
```

```
      "traits": {
        "smithy.api#output": { }
      }
    },
    "com.amazonaws.licensemanager#AccessDeniedException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {

"code": "ServiceAccessDenied",
          "httpResponseCode": 401
        },
        "smithy.api#documentation": "<p>Access to resource denied.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 401
      }
    },
    "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
      "type": "enum",
      "members": {
        "DISTRIBUTED_GRANTS_ONLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
          }
        },
        "ALL_GRANTS_PERMITTED_BY_ISSUER": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#AllowedOperation": {

"type": "enum",
      "members": {
        "CREATE_GRANT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateGrant"
          }
```

```json
        },
        "CHECKOUT_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutLicense"
          }
        },
        "CHECKOUT_BORROW_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutBorrowLicense"
          }
        },
        "CHECK_IN_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckInLicense"
          }
        },
        "EXTEND_CONSUMPTION_LICENSE":
{
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ExtendConsumptionLicense"
          }
        },
        "LIST_PURCHASED_LICENSES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ListPurchasedLicenses"
          }
        },
        "CREATE_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateToken"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#AllowedOperationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#AllowedOperation"
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
```

          "max":
7
        }
      }
    },
    "com.amazonaws.licensemanager#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 2048
        },
        "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
      }
    },
    "com.amazonaws.licensemanager#ArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
      }
    },
    "com.amazonaws.licensemanager#AuthorizationException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError":
{
          "code": "AuthorizationFailure",
          "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to perform the action. Check the IAM\n   policy associated with this account.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 403
      }
    },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
      "type": "structure",
      "members": {
        "LastRunTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"

```
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes automated discovery.</p>"
    }
  },

"com.amazonaws.licensemanager#Boolean": {
    "type": "boolean",
    "traits": {
      "smithy.api#default": false
    }
  },
  "com.amazonaws.licensemanager#BorrowConfiguration": {
    "type": "structure",
    "members": {
      "AllowEarlyCheckIn": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxTimeToLiveInMinutes": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
          "smithy.api#required": {}
        }
      }

},
    "traits": {
      "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#BoxBoolean": {
    "type": "boolean"
  },
  "com.amazonaws.licensemanager#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licensemanager#BoxLong": {
    "type": "long"
  },
  "com.amazonaws.licensemanager#CheckInLicense": {
    "type": "operation",
```

```
    "input": {
        "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
    }
},
"com.amazonaws.licensemanager#CheckInLicenseRequest":
{
    "type": "structure",
    "members": {
        "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>License consumption token.</p>",
                "smithy.api#required": {}
            }
```

```
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CheckInLicenseResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
          },
          {
            "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
          },
```

```
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
    }
  },
  "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
    "type": "structure",
    "members":
{
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
          "smithy.api#required": {}
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementDataList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
          "smithy.api#required": {}
        }
      },
      "DigitalSignatureMethod": {
        "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
        "traits": {
          "smithy.api#documentation": "<p>Digital signature method. The possible
value is JSON Web Signature (JWS) algorithm PS384.\n      For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
          "smithy.api#required": {}
        }
      },
```

```
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          },
          "CheckoutMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about constraints.</p>"
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>"
            }
          },
          "EntitlementsAllowed":
{
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
            }
          },
          "NodeId": {
```

```
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Node ID.</p>"
            }
        },
        "SignedToken": {
            "target": "com.amazonaws.licensemanager#SignedToken",
            "traits": {
                "smithy.api#documentation": "<p>Signed token.</p>"
            }
        },
        "IssuedAt": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
            }
        },
        "Expiration": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
            }
        },
        "CheckoutMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
                "smithy.api#documentation": "<p>Information about constraints.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#CheckoutLicense": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
```

```
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target":
"com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n            <p>If the
account that created the license is the same that is performing the check out, you must\n            specify the account
as the beneficiary.</p>\n        </note>"
    }
  },
  "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
    "type": "structure",
    "members": {
      "ProductSKU": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product
SKU.</p>",
          "smithy.api#required": {}
        }
      },
      "CheckoutType": {
        "target": "com.amazonaws.licensemanager#CheckoutType",
        "traits": {
```

```
                    "smithy.api#documentation": "<p>Checkout type.</p>",
                    "smithy.api#required": {}
                }
            },
            "KeyFingerprint": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
                    "smithy.api#required": {}
                }
            },
            "Entitlements": {
                "target": "com.amazonaws.licensemanager#EntitlementDataList",
                "traits": {
                    "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
                }
            },
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            },
            "Beneficiary": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License beneficiary.</p>"
                }
            },
            "NodeId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Node ID.</p>"
                }
            }
        },
        "traits": {

    "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
        "type": "structure",
        "members": {
```

```
"CheckoutType": {
    "target": "com.amazonaws.licensemanager#CheckoutType",
    "traits": {
        "smithy.api#documentation": "<p>Checkout type.</p>"
    }
},
"LicenseConsumptionToken": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>License consumption token.</p>"
    }
},
"EntitlementsAllowed": {
    "target": "com.amazonaws.licensemanager#EntitlementDataList",
    "traits": {
        "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
    }
},
"SignedToken": {
    "target": "com.amazonaws.licensemanager#SignedToken",
    "traits": {
        "smithy.api#documentation": "<p>Signed token.</p>"
    }
},
"NodeId": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>Node ID.</p>"
    }
},
"IssuedAt": {
    "target": "com.amazonaws.licensemanager#ISO8601DateTime",
    "traits": {
        "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
    }
},
"Expiration": {
    "target": "com.amazonaws.licensemanager#ISO8601DateTime",
    "traits": {
        "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
    }
},
"LicenseArn": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
    }
```

```
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutType": {
        "type": "enum",
        "members": {
          "PROVISIONAL": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PROVISIONAL"
            }
          },
          "PERPETUAL": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PERPETUAL"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#ClientRequestToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 36
          }
        }
      },
      "com.amazonaws.licensemanager#ClientToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 2048
          },
          "smithy.api#pattern": "^\\S+$"
        }
      },
      "com.amazonaws.licensemanager#ConflictException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
```

```
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "ConflictException",
              "httpResponseCode":
409
            },
            "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 409
          }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
          "type": "structure",
          "members": {
            "ResourceType": {
              "target": "com.amazonaws.licensemanager#ResourceType",
              "traits": {
                "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
              }
            },
            "ConsumedLicenses": {
              "target": "com.amazonaws.licensemanager#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
              }
            }
          },
      "traits": {
            "smithy.api#documentation": "<p>Details about license consumption.</p>"
          }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
          }
        },
        "com.amazonaws.licensemanager#ConsumptionConfiguration": {
          "type": "structure",
          "members": {
            "RenewType": {
              "target": "com.amazonaws.licensemanager#RenewType",
              "traits": {
                "smithy.api#documentation": "<p>Renewal frequency.</p>"
              }
```

```
        },
        "ProvisionalConfiguration": {
          "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
          }
        },
        "BorrowConfiguration": {
          "target": "com.amazonaws.licensemanager#BorrowConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target":
"com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
```

```
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
      },
    "com.amazonaws.licensemanager#CreateGrantRequest": {
        "type": "structure",
        "members": {
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "GrantName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant name.</p>",
              "smithy.api#required": {}

    }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Principals": {
          "target": "com.amazonaws.licensemanager#PrincipalArnList",
          "traits": {
            "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n        (ARN):</p>\n        <ul>\n          <li>\n            <p>An Amazon Web Services
account, which includes only the account specified.</p>\n          </li>\n        </ul>\n        <ul>\n          <li>\n
   <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n          </li>\n        </ul>\n
<ul>\n
         <li>\n            <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n        </ul>",
            "smithy.api#required": {}
          }
```

```
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },

    "com.amazonaws.licensemanager#CreateGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersion":
```

```
{
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
    }
},
"com.amazonaws.licensemanager#CreateGrantVersionRequest": {
    "type": "structure",
    "members": {
        "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
            }
```

```
        },
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant name.</p>"
        }
      },
      "AllowedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
        "traits": {
          "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#GrantStatus",
        "traits": {
          "smithy.api#documentation": "<p>Grant status.</p>"
        }
      },
      "StatusReason": {
        "target": "com.amazonaws.licensemanager#StatusReasonMessage",
        "traits": {
          "smithy.api#documentation": "<p>Grant status reason.</p>"
        }
      },
      "SourceVersion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Current version of the grant.</p>"
        }
      },
      "Options": {
        "target": "com.amazonaws.licensemanager#Options",
        "traits": {
          "smithy.api#documentation": "<p>The options specified for the grant.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}
```

```
            },
        "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant ARN.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>New version of the grant.</p>"
                    }
                }
            },
            "traits":
{

                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicense": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
           },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
           },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
```

all of these), license affinity to host (how long a \n   license must be associated with a host), and the number of
licenses purchased and used.</p>"
          }
        },
      "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the license configuration.</p>"
            }
          },
          "LicenseCountingType": {
            "target": "com.amazonaws.licensemanager#LicenseCountingType",
            "traits": {
              "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
            }
          },
          "LicenseCountHardLimit": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n   blocks the launch of new instances.</p>"
            }
          },
          "LicenseRules": {
            "target":
"com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n   vary by dimension, as follows.</p>\n
\n   \n   <p>\n   <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
    <code>licenseAffinityToHost</code> |\n   <code>maximumCores</code> |

<code>minimumCores</code>\n   </p>\n   \n   \n   <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n   <code>maximumCores</code> |
<code>minimumCores</code> |\n   <code>maximumSockets</code> | <code>minimumSockets</code> |\n
     <code>maximumVcpus</code> | <code>minimumVcpus</code>\n   </p>\n   \n
\n

 <p>\n   <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n   <code>maximumSockets</code> |
<code>minimumSockets</code>\n   </p>\n   \n   \n   <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n   <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n   </p>\n   \n   \n   <p>The unit for
<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for
<code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n
<code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n
<code>True</code> and <code>False</code>.</p>"
                    }
                },

   "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                    "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
                }
            },
            "DisassociateWhenNotFound": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            },
            "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                    "smithy.api#documentation": "<p>Product information.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",

```
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {

          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
```

```
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
                "smithy.api#required": {}
            }
        },
        "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
                "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n        For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
                "smithy.api#required": {}
            }
        },
        "DestinationLicenseContext":
{
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

        "traits": {
                "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
    "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "ReportGeneratorName":

{
          "target": "com.amazonaws.licensemanager#ReportGeneratorName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>",
            "smithy.api#required": {}
          }
        },
        "Type": {
```

                "target": "com.amazonaws.licensemanager#ReportTypeList",
                "traits": {
                    "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n   \n   \n   <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n   \n   \n   <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n   \n
",
                    "smithy.api#required": {}

                }
            },
            "ReportContext": {
                "target": "com.amazonaws.licensemanager#ReportContext",
                "traits": {
                    "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
                    "smithy.api#required": {}
                }
            },
            "ReportFrequency": {
                "target": "com.amazonaws.licensemanager#ReportFrequency",
                "traits": {
                    "smithy.api#documentation": "<p>Frequency by which reports are generated. Reports can be
generated daily, monthly, or weekly.</p>",
                    "smithy.api#required": {}
                }
            },
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientRequestToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique,
 case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            },
            "Description": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Description of the report generator.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                    "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
                }
            }
        },

```
          "traits": {
            "smithy.api#input": {}
          }
        },
      "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn":
 {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product
name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product SKU.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits": {
```

                     "smithy.api#documentation": "<p>License issuer.</p>",
                     "smithy.api#required": {}
                  }
               },
               "HomeRegion": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                     "smithy.api#documentation": "<p>Home Region for the license.</p>",
                     "smithy.api#required": {}

      }
               },
               "Validity": {
                  "target": "com.amazonaws.licensemanager#DatetimeRange",
                  "traits": {
                     "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
                     "smithy.api#required": {}
                  }
               },
               "Entitlements": {
                  "target": "com.amazonaws.licensemanager#EntitlementList",
                  "traits": {
                     "smithy.api#documentation": "<p>License entitlements.</p>",
                     "smithy.api#required": {}
                  }
               },
               "Beneficiary": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                     "smithy.api#documentation": "<p>License beneficiary.</p>",
                     "smithy.api#required": {}
                  }
                },
               "ConsumptionConfiguration": {
                  "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                  "traits": {
                     "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n   running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n   usage.</p>",
                     "smithy.api#required": {}
                  }
               },
               "LicenseMetadata": {
                  "target": "com.amazonaws.licensemanager#MetadataList",
                  "traits": {
                     "smithy.api#documentation": "<p>Information about the license.</p>"
                  }

```
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersion": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
      },
```

```
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#ConflictException"
       },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
    "type": "structure",
    "members": {
      "LicenseArn": {
        "target":
"com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License name.</p>",
          "smithy.api#required": {}
        }
      },
```

```
    "ProductName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product name.</p>",
        "smithy.api#required": {}
      }
    },
    "Issuer": {
      "target": "com.amazonaws.licensemanager#Issuer",
      "traits":
{

        "smithy.api#documentation": "<p>License issuer.</p>",
        "smithy.api#required": {}
      }
    },
    "HomeRegion": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Home Region of the license.</p>",
        "smithy.api#required": {}
      }
    },
    "Validity": {
      "target": "com.amazonaws.licensemanager#DatetimeRange",
      "traits": {
        "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
        "smithy.api#required": {}
      }
    },
    "LicenseMetadata": {
      "target": "com.amazonaws.licensemanager#MetadataList",
      "traits": {
        "smithy.api#documentation": "<p>Information about the license.</p>"
      }
    },
    "Entitlements": {
      "target": "com.amazonaws.licensemanager#EntitlementList",
      "traits": {
        "smithy.api#documentation": "<p>License entitlements.</p>",
        "smithy.api#required": {}
      }
    },
    "ConsumptionConfiguration": {
      "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
      "traits": {
        "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
```

workloads with offline\n   usage.</p>",
                "smithy.api#required": {}
            }
        },

    "Status": {
                "target": "com.amazonaws.licensemanager#LicenseStatus",
                "traits": {
                    "smithy.api#documentation": "<p>License status.</p>",
                    "smithy.api#required": {}
                }
            },
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            },
            "SourceVersion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Current version of the license.</p>"
                }
            }
        },
        "traits": {

    "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>License ARN.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>New version of the license.</p>"
                }
            },
            "Status": {

```
                "target": "com.amazonaws.licensemanager#LicenseStatus",
                "traits": {
                    "smithy.api#documentation": "<p>License status.</p>"
                }
            }
        }
    },
    "traits": {
        "smithy.api#output":
{}
    }
},
"com.amazonaws.licensemanager#CreateToken": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CreateTokenRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CreateTokenResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },

        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
```

token used to get an access token. With an access token,\n   you can call AssumeRoleWithWebIdentity to get role credentials that you can use to \n   call License Manager to manage the specified license.</p>"
```
                }
            },
            "com.amazonaws.licensemanager#CreateTokenRequest": {
                "type": "structure",
                "members": {
                    "LicenseArn": {
                        "target": "com.amazonaws.licensemanager#Arn",
                        "traits":
 {
                            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is mapped to the aud claim of the\n        JWT token.</p>",
                            "smithy.api#required": {}
                        }
                    },
                    "RoleArns": {
                        "target": "com.amazonaws.licensemanager#ArnList",
                        "traits": {
                            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the token. \n        License Manager does not check whether the roles are in use.</p>"
                        }
                    },
                    "ExpirationInDays": {
                        "target": "com.amazonaws.licensemanager#Integer",
                        "traits": {
                            "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The default is 365 days.</p>"
                        }
                    },
                    "TokenProperties": {
                        "target": "com.amazonaws.licensemanager#MaxSize3StringList",
                        "traits": {
                            "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The data is mapped\n        to the amr claim of the JWT token.</p>"
                        }
                    },
                    "ClientToken": {
                        "target": "com.amazonaws.licensemanager#ClientToken",
                        "traits": {
                            "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
                            "smithy.api#required": {}
                        }
                    }
                },
                "traits": {
                    "smithy.api#input": {}
                }
```

```
    },
    "com.amazonaws.licensemanager#CreateTokenResponse": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",


          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
        "TokenType": {
          "target": "com.amazonaws.licensemanager#TokenType",
          "traits": {
            "smithy.api#documentation": "<p>Token type.</p>"
          }
        },
        "Token": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DateTime": {
      "type": "timestamp"
    },
    "com.amazonaws.licensemanager#DatetimeRange": {
      "type": "structure",
      "members": {
        "Begin": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Start of the time range.</p>",
            "smithy.api#required": {}
          }
        },
        "End": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>End of the time range.</p>"
          }
        }
      },
```

```
      "traits": {
        "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteGrantRequest": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
```

```json
            }
          },
          "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the
grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation":
"<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
```

```
"com.amazonaws.licensemanager#DeleteLicense": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#RedirectException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes the specified license.</p>"
  }
},
"com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
  },
```

```
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
        "type": "structure",
        "members":
{
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>ID of the license configuration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
```

```
          "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
              "smithy.api#required": {}
            }
          }
```

```
        },
        "traits": {

"smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "SourceVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the license.</p>",
            "smithy.api#required":
{}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseResponse": {
      "type": "structure",
      "members": {
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "DeletionDate": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
```

```
              "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteToken": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },

          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteTokenRequest": {
        "type": "structure",
        "members": {
```

```
            "TokenId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token ID.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteTokenResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }

      }
    },
    "com.amazonaws.licensemanager#DigitalSignatureMethod": {
      "type": "enum",
      "members": {
        "JWT_PS384": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "JWT_PS384"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#Entitlement": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement name.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
          }
        },
```

```
      "MaxCount": {
        "target": "com.amazonaws.licensemanager#Long",
        "traits": {
          "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
        }
      },
      "Overage": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
        }
      },
      "Unit": {
        "target": "com.amazonaws.licensemanager#EntitlementUnit",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement unit.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowCheckIn": {
        "target":
"com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
    }
  },
  "com.amazonaws.licensemanager#EntitlementData": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement data name.</p>",
          "smithy.api#required": {}
        }
      },
      "Value": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement
data value.</p>"
        }
      },
```

```
      "Unit": {
        "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement data unit.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
    }
  },
  "com.amazonaws.licensemanager#EntitlementDataList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#EntitlementData"
    }
  },
  "com.amazonaws.licensemanager#EntitlementDataUnit": {
    "type": "enum",
    "members": {
      "COUNT": {
        "target": "smithy.api#Unit",
        "traits": {

"smithy.api#enumValue": "Count"
        }
      },
      "NONE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "None"
        }
      },
      "SECONDS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Seconds"
        }
      },
      "MICROSECONDS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Microseconds"
        }
      },
      "MILLISECONDS": {
        "target": "smithy.api#Unit",
```

```
        "traits": {
          "smithy.api#enumValue": "Milliseconds"
        }
      },
      "BYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes"
        }
      },
      "KILOBYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes"
        }
      },
      "MEGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes"
        }
      },
      "GIGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes"
        }
      },
      "TERABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabytes"
        }
      },
      "BITS":
{
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits"
        }
      },
      "KILOBITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits"
        }
      },
      "MEGABITS": {
```

```
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Megabits"
            }
        },
        "GIGABITS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Gigabits"
            }
        },
        "TERABITS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Terabits"

}
        },
        "PERCENT": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Percent"
            }
        },
        "BYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Bytes/Second"
            }
        },
        "KILOBYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Kilobytes/Second"
            }
        },
        "MEGABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Megabytes/Second"
            }
        },
        "GIGABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
             "traits": {
                "smithy.api#enumValue": "Gigabytes/Second"
            }
        },
```

```
          "TERABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Terabytes/Second"
            }
          },
          "BITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Bits/Second"
            }
          },
          "KILOBITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Kilobits/Second"
            }
          },
          "MEGABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue":
"Megabits/Second"
            }
          },
          "GIGABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Gigabits/Second"
            }
          },
          "TERABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Terabits/Second"
            }
          },
          "COUNT_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count/Second"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementList": {
      "type": "list",
      "member": {
```

```
            "target": "com.amazonaws.licensemanager#Entitlement"
        }
    },
    "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#EntitlementUnit": {
        "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "SECONDS":
{
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "Seconds"
              }
          },
          "MICROSECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Microseconds"
            }
          },
          "MILLISECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
```

```
          "smithy.api#enumValue": "Milliseconds"
        }
      },
      "BYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes"
        }
      },
      "KILOBYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes"
        }
      },
      "MEGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes"
        }
      },
      "GIGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes"
        }
      },
      "TERABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabytes"
        }
      },
      "BITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits"
        }
      },
      "KILOBITS": {
        "target": "smithy.api#Unit",
        "traits": {

          "smithy.api#enumValue": "Kilobits"
        }
      },
      "MEGABITS": {
        "target": "smithy.api#Unit",
```

```
        "traits": {
          "smithy.api#enumValue": "Megabits"
        }
      },
      "GIGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits"
        }
      },
      "TERABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits"
        }
      },
      "PERCENT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Percent"
        }
      },
      "BYTES_PER_SECOND": {
        "target":
"smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes/Second"
        }
      },
      "KILOBYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes/Second"
        }
      },
      "MEGABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes/Second"
        }
      },
      "GIGABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes/Second"
        }
      },
      "TERABYTES_PER_SECOND": {
```

```
                    "target": "smithy.api#Unit",
                    "traits": {

"smithy.api#enumValue": "Terabytes/Second"
                    }
                },
                "BITS_PER_SECOND": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Bits/Second"
                    }
                },
                "KILOBITS_PER_SECOND": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Kilobits/Second"
                    }
                },
                "MEGABITS_PER_SECOND": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Megabits/Second"
                    }
                },
                "GIGABITS_PER_SECOND": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Gigabits/Second"
                    }
                },

        "TERABITS_PER_SECOND": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Terabits/Second"
                    }
                },
                "COUNT_PER_SECOND": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Count/Second"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#EntitlementUsage": {
            "type": "structure",
            "members": {
```

```
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement usage name.</p>",
              "smithy.api#required": {}
            }
          },
          "ConsumedValue": {
            "target": "com.amazonaws.licensemanager#String",

          "traits": {
              "smithy.api#documentation": "<p>Resource usage consumed.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxCount": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList":
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementUsage"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
      },
```

```
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
    }
  },
  "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
    "type": "structure",
    "members": {
      "LicenseConsumptionToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License consumption token.</p>",
          "smithy.api#required": {}

      }
    },
      "DryRun": {
        "target": "com.amazonaws.licensemanager#Boolean",
        "traits": {
          "smithy.api#default": false,
          "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
        }
      }
    },
    "traits": {
```

```
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },

        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#FailedDependencyException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        },
        "ErrorCode": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "FailedDependency",
          "httpResponseCode": 424
        },
        "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 424
      }
    },
    "com.amazonaws.licensemanager#Filter": {
      "type": "structure",
      "members": {
```

```
      "Name": {
        "target": "com.amazonaws.licensemanager#FilterName",
        "traits": {
          "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanager#FilterValues",
        "traits": {
          "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter
name and value pair that is used to return more specific results from a\n    describe operation. Filters can be used
to match a set of resources by specific criteria,\n    such as tags, attributes, or IDs.</p>"
    }
  },
  "com.amazonaws.licensemanager#FilterLimitExceededException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "FilterLimitExceeded",
        "httpResponseCode": 400
      },
      "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Filter"
    }
  },
  "com.amazonaws.licensemanager#FilterName": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValue": {
    "type": "string"
```

```
        },
        "com.amazonaws.licensemanager#FilterValues": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#FilterValue",
                "traits": {
                    "smithy.api#xmlName": "item"
                }
            }
        },
        "com.amazonaws.licensemanager#Filters": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#Filter",
                "traits": {
                    "smithy.api#xmlName": "item"
                }
            }
        },
        "com.amazonaws.licensemanager#GetAccessToken": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
```

```
tokens\n        are valid for one hour.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetAccessTokenRequest": {
      "type": "structure",
      "members": {
        "Token": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
            "smithy.api#required": {}
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
            "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
      "type": "structure",
      "members": {
        "AccessToken": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Temporary access token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetGrantResponse"
      },
```

```
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
    }
  },
  "com.amazonaws.licensemanager#GetGrantRequest": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant version.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
```

```
        },
        "com.amazonaws.licensemanager#GetGrantResponse": {
            "type": "structure",
            "members": {
                "Grant": {
                    "target": "com.amazonaws.licensemanager#Grant",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant details.</p>"
                    }
                }
            },
            "traits": {

"smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicense": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetLicenseRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#GetLicenseResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"

},
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
            }
```

```
    },
    "com.amazonaws.licensemanager#GetLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits":
{
        "smithy.api#input": {}
      }
    },
```

"com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
    "type": "structure",
    "members": {
      "LicenseConfigurationId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
        }
      },
      "LicenseConfigurationArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
        }
      },
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Name of the license configuration.</p>"
        }
      },
      "Description": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Description of the license configuration.</p>"
        }
      },
      "LicenseCountingType": {
        "target": "com.amazonaws.licensemanager#LicenseCountingType",
        "traits": {
          "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",

        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",

```
          "traits": {
            "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License configuration status.</p>"
          }

        },
        "OwnerAccountId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
          }
        },
        "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
          }
        },
        "ManagedResourceSummaryList": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>Product information.</p>"
          }
        },
        "AutomatedDiscoveryInformation": {
```

```
            "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
            "traits": {
              "smithy.api#documentation": "<p>Automated discovery information.</p>"
            }
          },
          "DisassociateWhenNotFound": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTask": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {

            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
```

        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"

```
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
                    }
                },

    "StartTime": {
                    "target": "com.amazonaws.licensemanager#DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                    }
                },
                "LicenseConversionTime": {
                    "target": "com.amazonaws.licensemanager#DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                    }
                },
                "EndTime": {
                    "target": "com.amazonaws.licensemanager#DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
```

        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
            "smithy.api#required":
{}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
    "type": "structure",
    "members": {
      "ReportGenerator": {
        "target": "com.amazonaws.licensemanager#ReportGenerator",
        "traits": {
          "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license

```
configurations.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseRequest": {
        "type": "structure",
        "members": {
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                    "smithy.api#required": {}
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License version.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseResponse": {
        "type": "structure",
        "members": {
            "License": {
                "target": "com.amazonaws.licensemanager#License",
                "traits": {
                    "smithy.api#documentation": "<p>License details.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },

    "com.amazonaws.licensemanager#GetLicenseUsage": {
        "type": "operation",
        "input": {
```

          "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {

"target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
        "type": "structure",
        "members": {

```
            "LicenseUsage": {
                "target":
"com.amazonaws.licensemanager#LicenseUsage",
                "traits": {
                    "smithy.api#documentation": "<p>License usage details.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
        }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
```

```
      "S3BucketArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n       and so on.</p>"

 }
      },
      "SnsTopicArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
        }
      },
      "OrganizationConfiguration": {
        "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n       cross-account discovery.</p>"
        }
      },
      "EnableCrossAccountsDiscovery": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
        }

    },
      "LicenseManagerResourceShareArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n       provides member accounts with access to this share.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#Grant": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "GrantName": {

  "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant name.</p>",
              "smithy.api#required": {}
            }
          },
          "ParentArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Parent ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "GranteePrincipalArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {

  "smithy.api#documentation": "<p>The grantee principal ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region of the grant.</p>",
              "smithy.api#required": {}
            }
          },
          "GrantStatus": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>",
              "smithy.api#required": {}
            }
          },
          "StatusReason": {
```

```
              "target": "com.amazonaws.licensemanager#StatusReasonMessage",
              "traits": {
                "smithy.api#documentation": "<p>Grant status reason.</p>"
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>",
                "smithy.api#required": {}
              }
            },
            "GrantedOperations": {
              "target": "com.amazonaws.licensemanager#AllowedOperationList",
              "traits": {
                "smithy.api#documentation": "<p>Granted operations.</p>",
                "smithy.api#required": {}
              }
            },
            "Options": {
              "target": "com.amazonaws.licensemanager#Options",
              "traits": {
                "smithy.api#documentation": "<p>The options specified for the grant.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation":
"<p>Describes a grant.</p>"
          }
        },
        "com.amazonaws.licensemanager#GrantList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Grant"
          }
        },
        "com.amazonaws.licensemanager#GrantStatus": {
          "type": "enum",
          "members": {
            "PENDING_WORKFLOW": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "PENDING_WORKFLOW"
              }
            },
            "PENDING_ACCEPT": {
              "target": "smithy.api#Unit",
```

          "traits": {
            "smithy.api#enumValue": "PENDING_ACCEPT"
          }
        },
        "REJECTED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"

          }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits":
{
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }

```
            }
        },
        "com.amazonaws.licensemanager#GrantedLicense": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
                    }
                },
                "LicenseName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License name.</p>"
                    }

},
                "ProductName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Product name.</p>"
                    }
                },
                "ProductSKU": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Product SKU.</p>"
                    }
                },
                "Issuer": {
                    "target": "com.amazonaws.licensemanager#IssuerDetails",
                    "traits": {
                        "smithy.api#documentation": "<p>Granted license issuer.</p>"
                    }
                },
                "HomeRegion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#LicenseStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Granted license status.</p>"
                    }
```

```
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
ISO8601-UTC format.</p>"
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>"
          }
        },
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
          }
        },
        "LicenseMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Granted license metadata.</p>"
          }
        },
        "CreateTime": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

  }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Version of the granted license.</p>"
          }
        },
        "ReceivedMetadata": {
          "target": "com.amazonaws.licensemanager#ReceivedMetadata",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>Granted license received metadata.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
        }
    },
    "com.amazonaws.licensemanager#GrantedLicenseList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#GrantedLicense"
        }
    },
    "com.amazonaws.licensemanager#ISO8601DateTime":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-
1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
        }
    },
    "com.amazonaws.licensemanager#Integer": {
        "type": "integer"
    },
    "com.amazonaws.licensemanager#InvalidParameterValueException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "InvalidParameterValueProvided",
                "httpResponseCode": 400
            },
            "smithy.api#documentation":
"<p>One or more parameter values are not valid.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#InvalidResourceStateException": {
```

```
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidResourceState",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },

    "com.amazonaws.licensemanager#InventoryFilter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the filter.</p>",
              "smithy.api#required": {}
            }
          },
          "Condition": {
            "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
            "traits": {
              "smithy.api#documentation": "<p>Condition of the filter.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An inventory filter.</p>"
        }
      },
```

```
"com.amazonaws.licensemanager#InventoryFilterCondition": {
  "type": "enum",
  "members": {
    "EQUALS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EQUALS"
      }
    },
    "NOT_EQUALS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "NOT_EQUALS"
      }
    },
    "BEGINS_WITH": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "BEGINS_WITH"
      }
    },
    "CONTAINS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "CONTAINS"
      }
    }
  }
},
"com.amazonaws.licensemanager#InventoryFilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanager#InventoryFilter"
  }
},
"com.amazonaws.licensemanager#Issuer": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Issuer name.</p>",
        "smithy.api#required": {}
      }
    },
    "SignKey": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
```

```
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
 of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#IssuerDetails": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>"
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#License": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseName": {
```

```
          "target": "com.amazonaws.licensemanager#String",
          "traits": {

       "smithy.api#documentation": "<p>License name.</p>"
          }
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#IssuerDetails",
          "traits": {
            "smithy.api#documentation": "<p>License issuer.</p>"
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {

       "smithy.api#documentation": "<p>Home Region of the license.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
```

```
                    }
                },
                "Entitlements": {

    "target": "com.amazonaws.licensemanager#EntitlementList",
                    "traits": {
                        "smithy.api#documentation": "<p>License entitlements.</p>"
                    }
                },
                "ConsumptionConfiguration": {
                    "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
                    }
                },
                "LicenseMetadata": {
                    "target": "com.amazonaws.licensemanager#MetadataList",
                    "traits": {
                        "smithy.api#documentation": "<p>License metadata.</p>"
                    }
                },
                "CreateTime": {
                    "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>License
creation time.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License version.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
            }
        },
        "com.amazonaws.licensemanager#LicenseConfiguration": {
            "type": "structure",
            "members": {
                "LicenseConfigurationId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
                    }
                },
```

```
"LicenseConfigurationArn": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
    }
},
"Name": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>Name of the license configuration.</p>"
    }
},
"Description": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>Description of the license configuration.</p>"
    }
},
"LicenseCountingType": {
    "target": "com.amazonaws.licensemanager#LicenseCountingType",
    "traits": {
        "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
    }
},
"LicenseRules": {
    "target": "com.amazonaws.licensemanager#StringList",
    "traits": {
        "smithy.api#documentation": "<p>License rules.</p>"
    }
},
"LicenseCount": {
    "target": "com.amazonaws.licensemanager#BoxLong",
    "traits": {
        "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
    }
},
"LicenseCountHardLimit": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
    "traits": {
        "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
    }
},
"DisassociateWhenNotFound": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
    "traits": {
        "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
    }
```

```
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Status of the license configuration.</p>"
        }
      },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

    }
      },
      "ConsumedLicenseSummaryList": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
        }
      },
      "ManagedResourceSummaryList": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
        }
      },
      "ProductInformationList": {
        "target": "com.amazonaws.licensemanager#ProductInformationList",
        "traits": {
          "smithy.api#documentation": "<p>Product information.</p>"
        }
      },
      "AutomatedDiscoveryInformation": {

        "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
        "traits": {
          "smithy.api#documentation": "<p>Automated discovery information.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
```

agreement that can be\n   consumed and enforced by License Manager. Components include specifications for the license\n   type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n   Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n   must be associated with a host), and the number of licenses purchased and used.</p>"
                }
            },
            "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
                "type": "structure",
                "members": {
                    "ResourceArn":
 {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
                        }
                    },
                    "ResourceType": {
                        "target": "com.amazonaws.licensemanager#ResourceType",
                        "traits": {
                            "smithy.api#documentation": "<p>Type of server resource.</p>"
                        }
                    },
                    "ResourceOwnerId": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource consuming licenses.</p>"
                        }
                    },
                    "AssociationTime": {
                        "target": "com.amazonaws.licensemanager#DateTime",
                        "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
                        }
                    },
                    "AmiAssociationScope": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-account</code>.</p>"
                        }
                    }
                },
                "traits": {
                    "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
                }
            },

```
"com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
    }
},
"com.amazonaws.licensemanager#LicenseConfigurationStatus": {
    "type": "enum",
    "members":
{
        "AVAILABLE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "AVAILABLE"
            }
        },
        "DISABLED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "DISABLED"
            }
        }
    }
},
"com.amazonaws.licensemanager#LicenseConfigurationUsage": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
        },
        "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits":
{
                "smithy.api#documentation": "<p>Type of resource.</p>"
            }
        },
        "ResourceStatus": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Status of the resource.</p>"
            }
        },
        "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
          }
        },
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
          }
        },

    "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurations": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfiguration"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
      "members": {
        "UsageOperation": {
          "target": "com.amazonaws.licensemanager#UsageOperation",
          "traits": {
            "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n     </a>\n        </p>"
          }
```

```
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {

      "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the conversion task.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
```

```
                }
            },
            "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation":
 "<p>The time the conversion task was started at.</p>"
                }
            },
            "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskId": {
        "type": "string",
        "traits": {

        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
        "type": "enum",
        "members": {
            "IN_PROGRESS": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "IN_PROGRESS"
                }
            },
            "SUCCEEDED": {
```

```
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUCCEEDED"
          }
        },
        "FAILED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTasks":
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConversionTask"
      }
    },
    "com.amazonaws.licensemanager#LicenseCountingType": {
      "type": "enum",
      "members": {
        "VCPU": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "vCPU"
          }
        },
        "INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Instance"
          }
        },
        "CORE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Core"
          }
        },
        "SOCKET": {
          "target": "smithy.api#Unit",
           "traits": {
            "smithy.api#enumValue": "Socket"
          }
        }
      }
```

```
      },
      "com.amazonaws.licensemanager#LicenseDeletionStatus": {
        "type": "enum",
        "members": {
          "PENDING_DELETE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_DELETE"
            }
          },
          "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DELETED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#License"
        }
      },
      "com.amazonaws.licensemanager#LicenseOperationFailure": {

        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Resource type.</p>"
            }
          },
          "ErrorMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Error message.</p>"
            }
          },
          "FailureTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
```

```
                "traits": {

        "smithy.api#documentation": "<p>Failure time.</p>"
                }
            },
            "OperationName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Name of the operation.</p>"
                }
            },
            "ResourceOwnerId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
                }
            },
            "OperationRequestedBy": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
                }
            },
            "MetadataList": {

                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Reserved.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailureList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
        }
    },
    "com.amazonaws.licensemanager#LicenseSpecification": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

              "smithy.api#required": {}
              }
          },
          "AmiAssociationScope": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecifications": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseSpecification"
        }
      },
      "com.amazonaws.licensemanager#LicenseStatus": {
        "type": "enum",
        "members": {
          "AVAILABLE": {
            "target": "smithy.api#Unit",

            "traits": {
                "smithy.api#enumValue": "AVAILABLE"
            }
          },
          "PENDING_AVAILABLE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_AVAILABLE"
            }
          },
          "DEACTIVATED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DEACTIVATED"
            }
          },
          "SUSPENDED": {
            "target": "smithy.api#Unit",
```

```
              "traits": {
                "smithy.api#enumValue": "SUSPENDED"
              }
            },
          "EXPIRED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "EXPIRED"
              }
            },

        "PENDING_DELETE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "PENDING_DELETE"
              }
            },
          "DELETED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "DELETED"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#LicenseUsage": {
          "type": "structure",
          "members": {
            "EntitlementUsages": {
              "target": "com.amazonaws.licensemanager#EntitlementUsageList",
              "traits": {
                "smithy.api#documentation": "<p>License entitlement usages.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

          }
        },
        "com.amazonaws.licensemanager#LicenseUsageException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
```

```
      "aws.protocols#awsQueryError": {
         "code": "LicenseUsageFailure",
         "httpResponseCode": 412
      },
      "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 412
   }
},
"com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
   "type": "operation",
   "input": {
      "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
   },
   "output": {

"target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
   },
   "errors": [
      {
         "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
         "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
         "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
      },
      {
         "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
         "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
         "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
   ],
   "traits": {
      "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
   }
},
"com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
   "type": "structure",
```

```
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurationAssociations": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
          "traits": {
            "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
          }
        },
        "NextToken": {

"target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
```

```
"com.amazonaws.licensemanager#ListDistributedGrants": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
    }
},
"com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
    "type": "structure",
    "members": {
        "GrantArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {

                "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
```

"smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:</p>\n   \n   \n   <p>\n   <code>LicenseArn</code>\n   </p>\n   \n   \n   <p>\n   <code>GrantStatus</code>\n   </p>\n   \n   \n   <p>\n   <code>GranteePrincipalARN</code>\n   </p>\n   \n   \n   <p>\n   <code>ProductSKU</code>\n   </p>\n   \n   \n   <p>\n   <code>LicenseIssuerName</code>\n   </p>\n   \n   "
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
      "type": "structure",
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",

    "traits": {
            "smithy.api#documentation": "<p>Distributed grant details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {

            "type": "operation",
            "input": {
               "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
            },
            "output": {
               "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
            },
            "errors": [
               {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
               },
               {
                  "target": "com.amazonaws.licensemanager#AuthorizationException"
               },
               {
                  "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
               },
               {
                  "target": "com.amazonaws.licensemanager#RateLimitExceededException"
               },
               {
                  "target": "com.amazonaws.licensemanager#ServerInternalException"
               }
            ],
            "traits": {
               "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
            }
         },
         "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
            "type": "structure",
            "members": {
               "LicenseConfigurationArn": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                     "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
                     "smithy.api#required": {}
                  }
               },
               "MaxResults": {
                  "target": "com.amazonaws.licensemanager#BoxInteger",
                  "traits": {
                     "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                  }
               },
               "NextToken": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {

```
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
          "type": "structure",
          "members": {
            "LicenseOperationFailureList": {
              "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
              "traits": {
                "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
        },

    "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
```

```
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the license configurations
 for your account.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
            "type": "structure",
            "members": {
                "LicenseConfigurationArns": {
                    "target": "com.amazonaws.licensemanager#StringList",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanager#Filters",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n        Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n        Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n
<p>\n        <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n        Logical operators are <code>EQUALS</code> |
```

<code>NOT_EQUALS</code>.</p>\n
    \n   \n   <p>\n   <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n   Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   "
                }
              }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
        "type": "structure",
        "members": {
          "LicenseConfigurations": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurations",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",

            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {

```
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
      "type": "structure",
      "members": {
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#Filters",
          "traits": {
            "smithy.api#documentation": "<p>\n        Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n      </p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
      "type": "structure",
      "members": {
        "LicenseConversionTasks": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configuration tasks for your
```

```
account.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target":
"com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n        <ul>\n          <li>\n            <p>\n              <code>LicenseConfigurationArn</code>\n
</p>\n          </li>\n        </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits":
{
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
        "type": "structure",
        "members": {
          "ReportGenerators": {
            "target": "com.amazonaws.licensemanager#ReportGeneratorList",
            "traits": {
              "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }

      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
          "smithy.api#required": {}
        }
```

```
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
        "type": "structure",
        "members": {
            "LicenseSpecifications": {
                "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                "traits": {
                    "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseVersions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
        },
        "errors": [
```

```
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
    }
  }
},
"com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
  "type": "structure",
  "members": {
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanager#MaxSize100",
      "traits": {
        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
```

```
          "type": "structure",
          "members": {
            "Licenses": {
              "target": "com.amazonaws.licensemanager#LicenseList",
              "traits": {
                "smithy.api#documentation": "<p>License details.</p>"
              }
            },

"NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenses": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicensesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicensesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
```

```
      "traits": {
        "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },

    "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>Beneficiary</code>\n            </p>\n
      </li>\n          <li>\n            <p>\n              <code>ProductSKU</code>\n            </p>\n          </li>\n
    <li>\n            <p>\n              <code>Fingerprint</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>Status</code>\n            </p>\n          </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token
for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicensesResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#LicenseList",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>License details.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {

"smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedGrants": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
```

specifying the\n   recipient as this Amazon Web Services account, your organization, or an organizational unit\n (OU) to which this member account belongs.</p>"
                    }
                },
                "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
                    "type": "operation",
                    "input": {
                        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
                    },
                    "output": {
                        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
                    },
                    "errors": [
                        {
                            "target": "com.amazonaws.licensemanager#AccessDeniedException"
                        },
                        {
                            "target": "com.amazonaws.licensemanager#AuthorizationException"
                        },
                        {
                            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                        },
                        {
                            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                        },
                        {
                            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                        },
                        {
                            "target": "com.amazonaws.licensemanager#ServerInternalException"
                        },
                        {
                            "target": "com.amazonaws.licensemanager#ValidationException"
                        }
                    ],
                    "traits": {
                        "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
                    }
                },
                "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
                    "type": "structure",
                    "members": {
                        "LicenseArn": {
                            "target": "com.amazonaws.licensemanager#Arn",
                            "traits": {
                                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
                                "smithy.api#required": {}
```

```
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
      <ul>\n        <li>\n          <p>\n            <code>ParentArn</code>\n          </p>\n        </li>\n
<li>\n          <p>\n            <code>GranteePrincipalArn</code>\n          </p>\n        </li>\n      </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
        "type": "structure",
        "members": {
          "Grants": {
            "target": "com.amazonaws.licensemanager#GrantList",
            "traits": {
              "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
```

```
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
      "type": "structure",
      "members": {
        "GrantArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n    <ul>\n      <li>\n        <p>\n          <code>ProductSKU</code>\n        </p>\n
      </li>\n      <li>\n        <p>\n          <code>LicenseIssuerName</code>\n        </p>\n
</li>\n      <li>\n        <p>\n          <code>LicenseArn</code>\n        </p>\n      </li>\n
<li>\n        <p>\n          <code>GrantStatus</code>\n        </p>\n      </li>\n      <li>\n
<p>\n
      <code>GranterAccountId</code>\n        </p>\n      </li>\n    </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
      "type": "structure",
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",
          "traits": {
            "smithy.api#documentation": "<p>Received grant details.</p>"
          }
        },
```

```
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#ListReceivedLicenses": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists received licenses.</p>"
    }
},
"com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
```

```
{
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {

            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
    }
},
"com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>Beneficiary</code>\n            </p>\n
          </li>\n          <li>\n            <p>\n              <code>ProductSKU</code>\n
            </p>\n          </li>\n        </ul>"
            }
        },
```

```json
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
          "type": "structure",
          "members": {
            "Licenses": {
              "target": "com.amazonaws.licensemanager#GrantedLicenseList",
              "traits": {
                "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
          "type": "structure",
          "members": {
            "LicenseArns": {
              "target": "com.amazonaws.licensemanager#ArnList",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
              }
            },
```

```
        "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n    <ul>\n    <li>\n    <p>\n    <code>ProductSKU</code>\n    </p>\n
    </li>\n    <li>\n    <p>\n    <code>Status</code>\n    </p>\n    </li>\n
<li>\n    <p>\n    <code>Fingerprint</code>\n    </p>\n    </li>\n    <li>\n
<p>\n    <code>IssuerName</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Beneficiary</code>\n    </p>\n    </li>\n    </ul>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
    "type": "structure",
    "members": {
        "Licenses": {
            "target": "com.amazonaws.licensemanager#GrantedLicenseList",
            "traits": {
                "smithy.api#documentation": "<p>Received license details.</p>"
            }
        },

        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
```

        },
        "com.amazonaws.licensemanager#ListResourceInventory": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#FailedDependencyException"
                },
                {
                    "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
            "type": "structure",
            "members": {
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {

"smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanager#InventoryFilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical operators\n   are supported:</p>\n   \n   \n   <p>\n   <code>account_id</code> - The ID of the Amazon Web Services account that owns the resource.\n   Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>application_name</code> - The name of the application. \n   Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n   \n   \n   <p>\n   <code>license_included</code> - The type of license included. \n   Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n   Possible values are <code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-server-web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n   <p>\n   <code>platform</code> - The platform of the resource.\n   Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n   \n   \n   <p>\n   <code>resource_id</code> - The ID of the resource. \n   Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>tag:<key></code> - The key/value combination of a tag assigned \n   to the resource. Logical operators are <code>EQUALS</code> (single account) or\n   <code>EQUALS</code> | <code>NOT_EQUALS</code> (cross account).</p>\n   \n   "
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
        "type": "structure",
        "members": {
            "ResourceInventoryList": {
                "target": "com.amazonaws.licensemanager#ResourceInventoryList",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the resources.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {

```
            "smithy.api#output": { }
        }
    },
    "com.amazonaws.licensemanager#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

    "smithy.api#required": { }
                }
            }
        },
        "traits": {
            "smithy.api#input": { }
        }
```

```
        },
        "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
              "target": "com.amazonaws.licensemanager#TagList",
              "traits": {
                "smithy.api#documentation": "<p>Information about the tags.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListTokens": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListTokensRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListTokensResponse"
          },
          "errors":
[
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists your tokens.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListTokensRequest": {
          "type": "structure",
          "members": {
```

```
          "TokenIds": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {

              "smithy.api#documentation": "<p>Token IDs.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n        <li>\n            <p>\n                <code>LicenseArns</code>\n            </p>\n        </li>\n
</ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
          "Tokens": {
            "target": "com.amazonaws.licensemanager#TokenList",
            "traits": {
              "smithy.api#documentation": "<p>Received token details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
```

```
        },
        "traits": {
          "smithy.api#output": { }
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
 {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n        consumption details by resource at a selected point in time. Use this action to audit the\n        current
license consumption for any license inventory and configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
```

```
          "smithy.api#required": {}
        }
      },
       "MaxResults": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#Filters",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n       are supported:</p>\n      <ul>\n         <li>\n            <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
       Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n
 <li>\n            <p>\n                <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n         <li>\n            <p>\n
    <code>resourceAccount</code> - The ID of the account that owns the resource. \n          Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n      </ul>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
    "type": "structure",
    "members": {

"LicenseConfigurationUsageList": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
        "traits": {
          "smithy.api#documentation": "<p>Information about the license configurations.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
```

```
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#Location": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#Long": {
    "type": "long"
  },
  "com.amazonaws.licensemanager#ManagedResourceSummary": {
    "type": "structure",
    "members": {
      "ResourceType":
{

        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
        }
      },
      "AssociationCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
    }
  },
  "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
    }
  },
  "com.amazonaws.licensemanager#MaxSize100": {

"type": "integer",
    "traits": {
      "smithy.api#range": {
        "min": 1,
        "max": 100
```

```
            }
          }
        },
        "com.amazonaws.licensemanager#MaxSize3StringList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 3
            }
          }
        },
        "com.amazonaws.licensemanager#Message": {
          "type": "string"
        },
        "com.amazonaws.licensemanager#Metadata": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The key name.</p>"
              }
            },

        "Value": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
          }
        },
        "com.amazonaws.licensemanager#MetadataList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Metadata"
          }
        },
        "com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
          "type": "structure",
          "members": {
```

```
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#Options": {
    "type": "structure",
    "members": {
      "ActivationOverrideBehavior": {
        "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
        "traits": {
          "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n          Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n          <ul>\n            <li>\n                <p>As a license administrator, you can
optionally specify an\n                  <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
  </li>\n            <li>\n                <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n                  account in your
organization.</p>\n            </li>\n            <li>\n                <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n                  <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n                  are activating the grant.</p>\n            </li>\n          </ul>\n          <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n            <dd>\n                <p>Use this value to activate a grant
without replacing any member accounts\n                  active grants for the same product.</p>\n            </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n            <dd>\n                <p>Use this value to activate a
grant and disable other active grants in any\n                  member accounts
 for the same product. This action will also replace their\n                  previously activated grants with this activated
grant.</p>\n            </dd>\n          </dl>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n        override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager</a> in the <i>License Manager User Guide</i>.</p>"
    }
  },
  "com.amazonaws.licensemanager#OrganizationConfiguration": {
    "type": "structure",
    "members": {
      "EnableIntegration": {
        "target": "com.amazonaws.licensemanager#Boolean",
```

```
            "traits": {
                "smithy.api#default": false,
                "smithy.api#documentation":
"<p>Enables Organizations integration.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
    }
},
"com.amazonaws.licensemanager#PrincipalArnList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#Arn"
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 1
        }
    }
},
"com.amazonaws.licensemanager#ProductInformation": {
    "type": "structure",
    "members": {
        "ResourceType": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Resource type. The
possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                "smithy.api#required": {}
            }
        },
        "ProductInformationFilterList": {
            "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
            "traits": {
                "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n    <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n    <p>The following filters and are supported when
the resource type \n    is <code>SSM_MANAGED</code>:</p>\n
    <ul>\n      <li>\n        <p>\n          <code>Application Name</code> - The name of the
application.\n        Logical operator is <code>EQUALS</code>.</p>\n      </li>\n      <li>\n
<p>\n          <code>Application Publisher</code> - The publisher of the application.\n        Logical
```

operator is <code>EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>Application Version</code> - The version of the application.\n   Logical operator is <code>EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>Platform Name</code> - The name of the platform.\n   Logical operator is <code>EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>Platform Type</code> - The platform type.\n   Logical operator is <code>EQUALS</code>.</p>\n

          \n   \n   <p>\n   <code>Tag:key</code> - The key of a tag attached to an Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is <code>NOT_EQUALS</code>. The key for your tag must be appended to <code>Tag:</code> following the example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you are not using values for the key.\n   </p>\n   \n   \n   <p>\n   <code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from automated discovery.\n   Logical operator is <code>NOT_EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>License Included</code> - The type of license included.\n   Logical operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n   Possible values are:\n   <code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-server-web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n   <p>The following filters and logical operators are supported when the resource type\n   is <code>RDS</code>:</p>\n   \n   \n   <p>\n   <code>Engine Edition</code> - The edition of the database engine.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-se2</code>.</p>\n   \n   \n   <p>\n   <code>License Pack</code> - The license pack.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>data guard</code> | \n   <code>diagnostic pack sqlt</code> | \n   <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n   <code>olap</code>.</p>\n   \n   ",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#ProductInformationFilter": {
        "type": "structure",
        "members": {
          "ProductInformationFilterName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Filter name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductInformationFilterValue": {
            "target": "com.amazonaws.licensemanager#StringList",

```
      "traits": {
            "smithy.api#documentation": "<p>Filter value.</p>"
         }
      },
      "ProductInformationFilterComparator": {
         "target": "com.amazonaws.licensemanager#String",
         "traits": {
            "smithy.api#documentation": "<p>Logical operator.</p>",
            "smithy.api#required": {}
         }
      }
   },
   "traits": {
      "smithy.api#documentation": "<p>Describes product information filters.</p>"
   }
},
"com.amazonaws.licensemanager#ProductInformationFilterList": {
   "type": "list",
   "member": {
      "target": "com.amazonaws.licensemanager#ProductInformationFilter"
   }
},
"com.amazonaws.licensemanager#ProductInformationList": {
   "type": "list",
   "member": {
      "target": "com.amazonaws.licensemanager#ProductInformation"
   }
},
"com.amazonaws.licensemanager#ProvisionalConfiguration": {
   "type": "structure",
   "members": {
      "MaxTimeToLiveInMinutes": {
         "target": "com.amazonaws.licensemanager#BoxInteger",
         "traits": {
            "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
            "smithy.api#required": {}
         }
      }
   },
   "traits": {
      "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
   }
},
"com.amazonaws.licensemanager#RateLimitExceededException": {
   "type": "structure",
   "members": {
      "Message": {
         "target": "com.amazonaws.licensemanager#Message"
```

```
                    }
                },
                "traits": {
                    "aws.protocols#awsQueryError": {
                        "code": "RateLimitExceeded",
                        "httpResponseCode": 429
                    },
                    "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
                    "smithy.api#error": "client",
                    "smithy.api#httpError": 429
                }
            },
            "com.amazonaws.licensemanager#ReceivedMetadata": {
                "type": "structure",
                "members": {
                    "ReceivedStatus": {
                        "target": "com.amazonaws.licensemanager#ReceivedStatus",
                        "traits": {
                            "smithy.api#documentation": "<p>Received status.</p>"
                        }
                    },
                    "ReceivedStatusReason": {
                        "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                        "traits": {
                            "smithy.api#documentation": "<p>Received
status reason.</p>"
                        }
                    },
                    "AllowedOperations": {
                        "target": "com.amazonaws.licensemanager#AllowedOperationList",
                        "traits": {
                            "smithy.api#documentation": "<p>Allowed operations.</p>"
                        }
                    }
                },
                "traits": {
                    "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
                }
            },
            "com.amazonaws.licensemanager#ReceivedStatus": {
                "type": "enum",
                "members": {
                    "PENDING_WORKFLOW": {
                        "target": "smithy.api#Unit",
                        "traits": {
                            "smithy.api#enumValue": "PENDING_WORKFLOW"
                        }
```

```
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"PENDING_ACCEPT"
          }
        },
        "REJECTED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"
          }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
           "traits": {
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#RedirectException": {
      "type": "structure",
```

```
      "members": {
        "Location": {
          "target": "com.amazonaws.licensemanager#Location",
          "traits": {
            "smithy.api#httpHeader": "Location"
          }
        },
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 308
      }
    },
    "com.amazonaws.licensemanager#RejectGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#RejectGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#RejectGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
```

```
      ],
      "traits": {
        "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#RejectGrantRequest": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}

}
    },
    "com.amazonaws.licensemanager#RejectGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#RenewType":
```

```
{
    "type": "enum",
    "members": {
      "NONE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "None"
        }
      },
      "WEEKLY": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Weekly"
        }
      },
      "MONTHLY": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Monthly"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#ReportContext": {
    "type": "structure",
    "members": {
      "licenseConfigurationArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {

    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
    }
  },
  "com.amazonaws.licensemanager#ReportFrequency": {
    "type": "structure",
    "members": {
      "value": {
        "target": "com.amazonaws.licensemanager#Integer",
        "traits": {
          "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
```

```
            }
          },
          "period": {
            "target": "com.amazonaws.licensemanager#ReportFrequencyType",

            "traits": {
              "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
        }
      },
      "com.amazonaws.licensemanager#ReportFrequencyType": {
        "type": "enum",
        "members": {
          "DAY": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DAY"
            }
          },
          "WEEK": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "WEEK"
            }
          },
          "MONTH": {
            "target": "smithy.api#Unit",
            "traits":
{
              "smithy.api#enumValue": "MONTH"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#ReportGenerator": {
        "type": "structure",
        "members": {
          "ReportGeneratorName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the report generator.</p>"
            }
          },
```

```
          "ReportType": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
              "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
            }
          },
          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
            }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits": {
              "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
            }
          },
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
            }
          },
          "LastRunStatus": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
            }
          },
          "LastRunFailureReason":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
            }
          },
          "LastReportGenerationTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
            }
          },
          "ReportCreatorAccount": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
```

generator.</p>"
                }
            },
            "Description": {
                "target": "com.amazonaws.licensemanager#String",

    "traits": {
                    "smithy.api#documentation": "<p>Description of the report generator.</p>"
                }
            },
            "S3Location": {
                "target": "com.amazonaws.licensemanager#S3Location",
                "traits": {
                    "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
                }
            },
            "CreateTime": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Time the report was created.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                    "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
                }

    }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportGeneratorList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#ReportGenerator"
        }
    },
    "com.amazonaws.licensemanager#ReportGeneratorName": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }

```
        }
      },
      "com.amazonaws.licensemanager#ReportType": {
        "type": "enum",
        "members": {
          "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
            }

          },
          "LICENSE_CONFIGURATION_USAGE_REPORT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "LicenseConfigurationUsageReport"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#ReportTypeList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#ReportType"
        }
      },
      "com.amazonaws.licensemanager#ResourceInventory": {
        "type": "structure",
        "members": {
          "ResourceId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits":
{
              "smithy.api#documentation": "<p>Type of resource.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
```

```
"Platform": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>Platform of the resource.</p>"
    }
},
"PlatformVersion": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
    }
},
"ResourceOwningAccountId":
{
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>Details about a resource.</p>"
}
},
"com.amazonaws.licensemanager#ResourceInventoryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#ResourceInventory"
    }
},
"com.amazonaws.licensemanager#ResourceLimitExceededException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
    },
    "traits": {
        "aws.protocols#awsQueryError": {
            "code": "ResourceLimitExceeded",

            "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
    }
},
```

```
"com.amazonaws.licensemanager#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "Message": {
      "target": "com.amazonaws.licensemanager#Message"
    }
  },
  "traits": {
    "aws.protocols#awsQueryError": {
      "code": "InvalidResource.NotFound",
      "httpResponseCode": 400
    },
    "smithy.api#documentation": "<p>The resource cannot be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 400
  }
},
"com.amazonaws.licensemanager#ResourceType": {
  "type": "enum",
  "members": {
    "EC2_INSTANCE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EC2_INSTANCE"
      }
    },
    "EC2_HOST": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EC2_HOST"
      }
    },
    "EC2_AMI": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EC2_AMI"
      }
    },
    "RDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "RDS"
      }
    },
    "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
      "target": "smithy.api#Unit",
      "traits": {
```

```
              "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#S3Location": {
        "type": "structure",
        "members": {
          "bucket": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
            }
          },
          "keyPrefix": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
        }
      },
      "com.amazonaws.licensemanager#ServerInternalException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InternalError",
            "httpResponseCode": 500
          },
          "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanager#SignedToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 4096
```

```
        }
      }
    },
    "com.amazonaws.licensemanager#StatusReasonMessage": {
      "type": "string",
      "traits":
{
        "smithy.api#length": {
          "min": 0,
          "max": 400
        },
        "smithy.api#pattern": "^[\\s\\S]+$"
      }
    },
    "com.amazonaws.licensemanager#String": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#StringList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#String"
      }
    },
    "com.amazonaws.licensemanager#Tag": {
      "type": "structure",
      "members": {
        "Key": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Tag key.</p>"
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation":
"<p>Tag value.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#TagKeyList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#String"
```

```
        }
      },
      "com.amazonaws.licensemanager#TagList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Tag"
        }
      },
      "com.amazonaws.licensemanager#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#TagResourceResponse"
        },
        "errors": [
          {
            "target":
"com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits":
{
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
```

```
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>One or more tags.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#TokenData": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>"
            }
          },
          "TokenType": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "ExpirationTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
```

```
      format.</p>"
            }
          },
          "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
            "traits": {
              "smithy.api#documentation": "<p>Data specified by the caller.</p>"
            }
          },
          "RoleArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Describes a token.</p>"
        }
      },
      "com.amazonaws.licensemanager#TokenList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#TokenData"
        }
      },
      "com.amazonaws.licensemanager#TokenString": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 4096
          },
          "smithy.api#pattern": "^\\S+$"
        }
      },
      "com.amazonaws.licensemanager#TokenType": {
        "type": "enum",
        "members": {
```

```
            "REFRESH_TOKEN": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "REFRESH_TOKEN"
              }
            }
          }
        },
    "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
  {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
```

```
        "traits": {
          "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#UntagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "TagKeys": {
            "target": "com.amazonaws.licensemanager#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
        },
        "output": {

 "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
```

```json
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseConfigurationStatus": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
            "traits": {
              "smithy.api#documentation": "<p>New status of the license configuration.</p>"
            }
          },
          "LicenseRules": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {

  "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
configuration is licenseAffinityToHost.</p>"
            }
          },
          "LicenseCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
```

```
              "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
            }
          },
          "LicenseCountHardLimit": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
            }
          },
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New description of the license configuration.</p>"
            }
          },
          "ProductInformationList": {
            "target": "com.amazonaws.licensemanager#ProductInformationList",
            "traits": {
              "smithy.api#documentation": "<p>New product information.</p>"
            }
          },
          "DisassociateWhenNotFound": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
            }
          }
        },
        "traits":
{
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
```

```
      },
   "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
      "type": "operation",
      "input": {
         "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
      },
      "output": {
         "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
      },
      "errors": [
         {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
         },
         {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
         },
         {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
         {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
         },
         {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
         },
         {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
         },
         {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
         },
         {
            "target": "com.amazonaws.licensemanager#ValidationException"
         }
      ],
      "traits": {
         "smithy.api#documentation": "<p>Updates a report generator.</p>\n         <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
      }
   },
   "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
      "type": "structure",
      "members": {
         "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
               "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
```

update.</p>",
                "smithy.api#required": {}
              }
            },
            "ReportGeneratorName": {
              "target": "com.amazonaws.licensemanager#ReportGeneratorName",
              "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>",
                "smithy.api#required": {}
              }
            },
            "Type": {
              "target": "com.amazonaws.licensemanager#ReportTypeList",
              "traits": {
                "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n   \n   \n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n   \n   \n   <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n   \n   ",
                "smithy.api#required": {}
              }
            },
            "ReportContext": {
              "target": "com.amazonaws.licensemanager#ReportContext",
              "traits": {
                "smithy.api#documentation": "<p>The report context.</p>",
                "smithy.api#required": {}
              }
            },
            "ReportFrequency": {
              "target": "com.amazonaws.licensemanager#ReportFrequency",
              "traits":
 {
                "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                "smithy.api#required": {}
              }
            },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientRequestToken",
              "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {

```
                    "smithy.api#documentation": "<p>Description of the report generator.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },

            {
                "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
            },
            {
                "target": "com.amazonaws.licensemanager#LicenseUsageException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
        }
    },

    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "AddLicenseSpecifications": {
                "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                "traits": {
                    "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
                }
            },
            "RemoveLicenseSpecifications": {
                "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                "traits": {
                    "smithy.api#documentation":
 "<p>ARNs of the license configurations to remove.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "S3BucketArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
            }
          },
          "SnsTopicArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
            }
          },
          "OrganizationConfiguration": {
            "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
```

```
                    }
                },
                "EnableCrossAccountsDiscovery": {
                    "target": "com.amazonaws.licensemanager#BoxBoolean",
                    "traits": {
                        "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#UsageOperation": {
            "type": "string",
            "traits": {
                "smithy.api#length": {
                    "min": 0,
                    "max": 50
                }
            }
        },
        "com.amazonaws.licensemanager#ValidationException": {
            "type": "structure",
            "members": {
                "Message": {
                    "target": "com.amazonaws.licensemanager#Message"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

     "smithy.api#error": "client",
                "smithy.api#httpError": 400
            }
        }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```go
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version
 // of the license.
 //
 // This member is required.
 SourceVersion *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {
```

```go
    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack,
options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
        return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
```

```go
    return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
}
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
```

```
  Region:      region,
  ServiceID:    ServiceID,
  OperationName: "DeleteLicense",
 }
}
```

1.24 aws-smithy-go 1.18.1

1.24.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

                       Apache License
                  Version 2.0, January 2004
                http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

1.25 aws-aws-sdk-go-v2 1.18.1
1.25.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

```go
// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out
 := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```
noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
 }
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if
err != nil {
 return err
 }
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
if err = addHTTPSignerV4Middleware(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
if err = addClientUserAgent(stack); err != nil {
 return err
 }
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
```

```go
err != nil {
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  SigningName:  "grafana",
  OperationName: "AssociateLicense",
 }
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

name: License Scan

```yaml
on: [pull_request]

jobs:
 licensescan:
  name: License Scan
  runs-on: ubuntu-latest
  strategy:
   matrix:
    python-version: [3.9]
```

```yaml
steps:
  - name: Checkout target
    uses: actions/checkout@v2
    with:
      path: sdkbase
      ref: ${{ github.base_ref }}
  - name: Checkout this ref
    uses: actions/checkout@v2
    with:
      path: new-ref
      fetch-depth: 0
  - name: Get Diff
    run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }} ${{ github.sha }} > refDiffFiles.txt
  - name: Get Target Files
    run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
  - name: Checkout scancode
    uses: actions/checkout@v2
    with:
      repository: nexB/scancode-toolkit
      path: scancode-toolkit
      fetch-depth: 1
  - name: Set up Python ${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
    "context"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```go
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary
    *string

    noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
```

```go
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
	return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
	return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
	return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
}
if err = addRetryMiddlewares(stack,
options); err != nil {
	return err
}
if err = addHTTPSignerV4Middleware(stack, options); err != nil {
	return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
	return err
}
if err = addClientUserAgent(stack); err != nil {
	return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
	return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
	return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
	return err
```

```go
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
   return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
   return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil
  {
   return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
   return err
  }
  return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  SigningName:   "license-manager",
  OperationName: "CheckInLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }
```

```go
	result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*CheckoutLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type
CheckoutLicenseInput struct {

	// Checkout type.
	//
	// This member is required.
	CheckoutType types.CheckoutType

	// Unique, case-sensitive identifier that you provide to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// License entitlements.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Key fingerprint identifying the license.
	//
	// This member is required.
	KeyFingerprint *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// License beneficiary.
	Beneficiary *string

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}
```

```go
type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name
 (ARN) of the checkout license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
```

```
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return
err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
	return err
	}
	if err = addHTTPSignerV4Middleware(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
	return err
	}
	if err = addClientUserAgent(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
	return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
	return err
	}
	if err = addRequestIDRetrieverMiddleware(stack);
err != nil {
	return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
	return err
```

```
            }
            if err = addRequestResponseLogging(stack, options); err != nil {
             return err
            }
            return nil
           }


           func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
           {
            return &awsmiddleware.RegisterServiceMetadata{
             Region:        region,
             ServiceID:      ServiceID,
             SigningName:    "license-manager",
             OperationName: "CheckoutLicense",
            }
           }
```

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
          }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
```

```
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {

            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
```

```
"LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    }
},
"LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n        Organizations integration has been enabled, and
which Regions data will be aggregated from.</p>"
    }
},
"Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
},
"StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
},
"HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays
the aggregated data for Linux\n        subscriptions.</p>"
    }
}
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }

        },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        },
        "Region": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
```

```
          "traits": {
             "smithy.api#documentation": "<p>The time in which the last discovery
updated the instance details.</p>"
          }
       },
       "SubscriptionName": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
          }
       }
    },
    "traits": {
       "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
       "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
    "type": "structure",
    "members": {
       "message": {
          "target": "smithy.api#String"
       }
    },
    "traits": {
       "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
       "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
       },
       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
       },
       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
       },
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
            }
        ],

    "traits": {
        "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
            "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                "x-api-key",
                "authorization",
                "x-amz-date",

    "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
            ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
                "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
```

```
                    "documentation": "The
 AWS region used to dispatch the request.",
                    "type": "String"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

                "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }
            },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {

                        "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
```

```
                    {
                       "ref": "UseFIPS"
                    },
                    true
                 ]
              }
           ],
           "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
           "type": "error"
        },
        {
           "conditions": [],
           "type": "tree",
           "rules": [
              {

        "conditions": [
                    {
                       "fn": "booleanEquals",
                       "argv": [
                          {
                             "ref": "UseDualStack"
                          },
                          true
                       ]
                    }
                 ],
                 "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                 "type": "error"
              },
              {
                 "conditions": [],

     "endpoint": {
                    "url": {
                       "ref": "Endpoint"
                    },
                    "properties": {},
                    "headers": {}
                 },
                 "type": "endpoint"
              }
           ]
        }
     ]
  },
  {
     "conditions": [],
```

```
"type": "tree",
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",

        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "type": "tree",
    "rules": [
      {
        "conditions": [
          {
            "fn": "aws.partition",
            "argv": [
              {
                "ref": "Region"
              }
            ],

            "assign": "PartitionResult"
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              },

              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
```

```
            },
            true
          ]
        }
      ],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {

              "fn": "booleanEquals",
              "argv": [
                true,
                {
                  "fn": "getAttr",
                  "argv": [
                    {
                      "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                  ]

}

              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                true,
                {
                  "fn": "getAttr",
                  "argv": [
                    {
                      "ref": "PartitionResult"
                    },

              "supportsDualStack"
                  ]
                }
              ]
            }
          ],
          "type": "tree",
          "rules": [
            {
              "conditions": [],
              "type": "tree",
```

"rules": [
          {

"conditions": [],
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
          }
        ]
    },
    {
      "conditions": [],
      "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
      "type": "error"
    }
  ]
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {

"ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "type": "tree",
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,

```
                          {
                                        "fn": "getAttr",
                                        "argv": [
                                           {
                                              "ref": "PartitionResult"
                                           },
                                           "supportsFIPS"
                                        ]
                                     }
                                  ]
                               }
                            ],
                            "type": "tree",

                      "rules": [
                               {
                                  "conditions": [],
                                  "type": "tree",
                                  "rules": [
                                     {
                                        "conditions": [],
                                        "endpoint": {
                                           "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                                           "properties": {},
                                           "headers": {}

                                        },
                                        "type": "endpoint"
                                     }
                                  ]
                               }
                            ]
                         },
                         {
                            "conditions": [],
                            "error": "FIPS is enabled but this partition does not support FIPS",
                            "type": "error"
                         }
                      ]

               },
                         {
                            "conditions": [
                               {
                                  "fn": "booleanEquals",
                                  "argv": [
                                     {
```

```json
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "type": "tree",
            "rules": [
              {

                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [],
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [],

                  "endpoint": {
                          "url": "https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ]
                  }
                ]
              },
```

```
                {
                                        "conditions": [],
                                        "error": "DualStack is enabled but this partition does not support DualStack",
                                        "type": "error"
                                    }
                                ]
                            },
                            {
                                "conditions": [],
                                "type": "tree",
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
            }
        ]
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                }
            },
            "params": {
                "UseDualStack": true,
```

```
          "UseFIPS": true,
          "Region": "us-gov-east-1"
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {

          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "UseDualStack": false,
          "UseFIPS": true,
          "Region": "us-gov-east-1"
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "UseDualStack": true,

"UseFIPS": false,
          "Region": "us-gov-east-1"
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "UseDualStack": false,
          "UseFIPS": false,
          "Region": "us-gov-east-1"
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
```

```
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "UseDualStack": true,
                "UseFIPS": true,
                "Region": "cn-north-1"
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                }
            },
            "params": {
                "UseDualStack": false,

                "UseFIPS": true,
                "Region": "cn-north-1"
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "UseDualStack": true,
                "UseFIPS": false,
                "Region": "cn-north-1"
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect":
{
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                }
            },
            "params": {
```

```
                            "UseDualStack": false,
                            "UseFIPS": false,
                            "Region": "cn-north-1"
                        }
                    },
                    {
                        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                        "expect": {
                            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                        },
                        "params": {
                            "UseDualStack": true,
                            "UseFIPS": true,

    "Region": "us-iso-east-1"
                        }
                    },
                    {
                        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                            }
                        },
                        "params": {
                            "UseDualStack": false,
                            "UseFIPS": true,
                            "Region": "us-iso-east-1"
                        }
                    },
                    {
                        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                        "expect": {
                            "error": "DualStack is enabled but
this partition does not support DualStack"
                        },
                        "params": {
                            "UseDualStack": true,
                            "UseFIPS": false,
                            "Region": "us-iso-east-1"
                        }
                    },
                    {
                        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
                            }
```

```
        },
      "params": {
        "UseDualStack": false,
        "UseFIPS": false,
        "Region": "us-iso-east-1"
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "UseDualStack": true,
        "UseFIPS": true,
        "Region": "us-east-1"
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"

      }
    },
      "params": {
        "UseDualStack": false,
        "UseFIPS": true,
        "Region": "us-east-1"
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "UseDualStack": true,
        "UseFIPS": false,
        "Region": "us-east-1"
      }
    },
    {
```

```
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
                      }
                    },
                    "params": {
                      "UseDualStack": false,
                      "UseFIPS": false,
                      "Region": "us-east-1"
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                    },
                    "params": {

                   "UseDualStack": true,
                      "UseFIPS": true,
                      "Region": "us-isob-east-1"
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                      }
                    },
                    "params": {
                      "UseDualStack": false,
                      "UseFIPS": true,
                      "Region": "us-isob-east-1"
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
                    "expect": {
                      "error": "DualStack is enabled but this partition does not support DualStack"
                    },
                    "params": {
                      "UseDualStack": true,
                      "UseFIPS": false,
                      "Region": "us-isob-east-1"
```

```
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                        }
                    },
                    "params": {
                        "UseDualStack": false,

"UseFIPS": false,
                        "Region": "us-isob-east-1"
                    }
                },
                {
                    "documentation": "For custom endpoint with fips disabled and dualstack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://example.com"
                        }
                    },
                    "params": {
                        "UseDualStack": false,
                        "UseFIPS": false,
                        "Region": "us-east-1",
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                    "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                    },
                    "params": {
                        "UseDualStack": false,
                        "UseFIPS": true,
                        "Region": "us-east-1",
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
```

```
                    "params": {
                        "UseDualStack": true,
                        "UseFIPS": false,
                        "Region": "us-east-1",

    "Endpoint": "https://example.com"
                        }
                    }
                ],
                "version": "1.0"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled LinuxSubscriptionsDiscovery"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
        "type": "structure",
        "members": {

            "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "OrganizationIntegration": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
                "traits": {
                    "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                    "smithy.api#required": {}
```

```
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with commercial Linux\n
    subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Instances"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
```

"Filters": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
  "traits": {
    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n
  more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>AmiID</code> with an optional operator to see subscriptions that match, partially\n   match, or don't match
a certain Amazon Machine Image (AMI) ID.</p>\n   <p>The valid names for this filter are:</p>\n   \n
    \n   <p>\n   <code>AmiID</code>\n   </p>\n   \n   \n
<p>\n   <code>InstanceID</code>\n   </p>\n   \n   \n   <p>\n
<code>AccountID</code>\n   </p>\n   \n   \n   <p>\n
<code>Status</code>\n   </p>\n   \n   \n   <p>\n
<code>Region</code>\n   </p>\n   \n   \n   <p>\n
<code>UsageOperation</code>\n
        </p>\n   \n   \n   <p>\n   <code>ProductCode</code>\n
</p>\n   \n   \n   <p>\n   <code>InstanceType</code>\n   </p>\n
\n   \n   <p>The valid Operators for this filter are:</p>\n   \n   \n   <p>\n
    <code>contains</code>\n   </p>\n   \n   \n   <p>\n
<code>equals</code>\n   </p>\n   \n   \n   <p>\n
<code>Notequal</code>\n   </p>\n   \n   "
  }
},
"MaxResults": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
  "traits": {
    "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Token for the next set of results.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 16384
    }
  }
}
},
"traits": {
  "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {

```
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n      organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
```

```
                    "items": "Subscriptions"
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
            "type": "structure",
            "members": {
                "Filters": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n        <p>The valid names for this filter are:</p>\n        <ul>\n
<li>\n            <p>\n                <code>Subscription</code>\n            </p>\n          </li>\n
</ul>\n        <p>The valid Operators for this filter are:</p>\n        <ul>\n          <li>\n              <p>\n
<code>contains</code>\n            </p>\n          </li>\n          <li>\n              <p>\n
<code>equals</code>\n            </p>\n          </li>\n          <li>\n              <p>\n
<code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the
next set of results.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
```

```
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",

"traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
              },
              {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
              },
              {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
              }
            ],
            "smithy.api#length": {
              "min": 1,
              "max": 20
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
```

```
          },
          {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "InProgress",
            "value": "InProgress",
            "documentation": "InProgress status"
          },
          {
            "name": "Completed",
            "value": "Completed",
            "documentation": "Completed status"
          },
          {
            "name": "Successful",
            "value": "Successful",
            "documentation": "Successful status"
          },
          {
            "name": "Failed",
            "value": "Failed",
            "documentation": "Failed status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
      "type": "list",
      "member":
{
        "target": "smithy.api#String",
        "traits": {
```

```
          "smithy.api#length": {
            "min": 1,
            "max": 100
          }
        }
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {

  "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
```

```
      },
      "traits": {

  "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
```

```
          "uri": "/subscription/UpdateServiceSettings"
        },

    "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
            "smithy.api#required": {}
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled,
 and which Regions data will be aggregated from.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowUpdate": {
          "target": "smithy.api#Boolean",
          "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n     allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled
 for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
```

                        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n   integration has been enabled, and which Regions data will be aggregated
from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n   subscriptions.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error":
 "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (

```go
	"context"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/grafana/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
	if params == nil {
		params = &DisassociateLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*DisassociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type DisassociateLicenseInput struct {

	// The type of license to remove from the workspace.
	//
	// This
	member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to remove the Grafana Enterprise license from.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

	// A structure containing information about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription
```

```go
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = addHTTPSignerV4Middleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack); err != nil {
		return err
	}
```

```go
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
		middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		SigningName:   "grafana",
		OperationName: "DisassociateLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
```

```go
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
if params == nil {
 params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version
 *string

 noSmithyDocumentSerde
}

type GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
```

```
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if
	err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = addHTTPSignerV4Middleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
	err != nil {
```

```go
        return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
        return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
        return err
    }
    if err = addResponseErrorMiddleware(stack);
    err != nil {
        return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:        region,
        ServiceID:     ServiceID,
        SigningName:   "license-manager",
        OperationName: "GetLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options)) (*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
    c.addOperationListLicensesMiddlewares)
```

```go
	if err != nil {
		return nil, err
	}

	out := result.(*ListLicensesOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type ListLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//   - Beneficiary
	//   - ProductSKU
	//   - Fingerprint
	//
	//   - Status
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListLicensesOutput struct {

	// License details.
	Licenses []types.License

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
	if err != nil {
```

```go
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
    return err
}
if err = addHTTPSignerV4Middleware(stack, options); err != nil {
    return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
}
if err = addClientUserAgent(stack); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
```

```go
    return err
   }
   if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
   }
   if err = addResponseErrorMiddleware(stack); err != nil {
    return err
   }
   if err = addRequestResponseLogging(stack, options); err != nil {
    return err
   }
   return nil
  }

  func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
   return &awsmiddleware.RegisterServiceMetadata{
    Region:       region,
    ServiceID:     ServiceID,
    SigningName:  "license-manager",
    OperationName: "ListLicenses",
   }
  }
```

```json
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
```

```
            "smithy.api#documentation":
"<p>Details about an Active Directory identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related
costs will take 48\n      hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
    monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
```

```
      "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n    resource.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
```

```
            "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/DeregisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
```

```json
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },

      "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
        },
        "Value": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used
to return more specific results from a describe\n      operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n      attributes, or IDs.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
        "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Settings": {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n      settings such as the subnets to provision VPC endpoints.</p>",
            "smithy.api#required":
 {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
```

```
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status
of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
                    "smithy.api#required": {}
                }
            },
            "LastStatusCheckDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
```

```
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits":
{
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
          "smithy.api#required": {}
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the user.</p>"
        }
      },
      "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
      },
      "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
```

```
                    "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
                }
            }

    },
        "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",

    "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
```

```
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    }

  },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
     },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
      ],
      "additionalExposedHeaders": [
```

```
                    "x-amzn-errortype",
                    "x-amzn-requestid",
                    "x-amzn-trace-id"


    ]
          },
          "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager User Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack
  endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "Boolean"
              },
              "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "Boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "String"
              }

        },
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
```

```
                              {
                                "ref": "Endpoint"
                              }
                          ]
                      }
                  ],
                  "type": "tree",
                  "rules": [
                      {
                          "conditions": [
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },

                                    true
                                  ]
                              }
                          ],
                          "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                          "type": "error"
                      },
                      {
                          "conditions": [],
                          "type": "tree",
                          "rules": [
                              {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
                                            {
                                                "ref": "UseDualStack"

                                            },
                                                true
                                            ]
                                    }
                                  ],
                                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                                  "type": "error"
                              },
                              {
                                "conditions": [],
                                "endpoint": {
                                    "url": {
```

```json
                            "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
                    }
                },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    },
    {
        "conditions": [],
        "type": "tree",
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Region"
                            }
                        ]
                    }
                ],
                "type": "tree",

    "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "aws.partition",
                                "argv": [
                                    {
                                        "ref": "Region"
                                    }
                                ],
                                "assign": "PartitionResult"
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {
                                "conditions": [
```

```json
                    {
                            "fn": "booleanEquals",
                            "argv": [
                              {
                                "ref": "UseFIPS"
                              },
                              true
                            ]
                          },
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              {
                                "ref": "UseDualStack"

          },
                              true
                            ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
                          {
                            "conditions": [
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  true,
                                  {

    "fn": "getAttr",

                                    "argv": [
                                      {
                                        "ref": "PartitionResult"
                                      },
                                      "supportsFIPS"
                                    ]
                                  }
                                ]
                              },
                              {
                                "fn": "booleanEquals",
                                "argv": [

                                  true,
                                  {
                                    "fn": "getAttr",
                                    "argv": [
```

```
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [],
                        "type": "tree",
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",

                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ]
                    }
                ]
            },
            {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

    "type": "error"
            }
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
```

```
                              true
                            ]
                        }
                    ],

            "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"

                                            },
                                                "supportsFIPS"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {
                                "conditions": [],
                                "type": "tree",
                                "rules": [

                                    {
                                            "conditions": [],
                                            "endpoint": {
                                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                                                "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                    }
                                ]
                            },
```

```
                                {
                                   "conditions": [],
                                   "error": "FIPS is enabled but this partition does not support FIPS",
                                   "type": "error"
                                }
                             ]
                          },
                          {
                             "conditions": [
                                {
                                   "fn": "booleanEquals",
                                   "argv": [

      {
                                         "ref": "UseDualStack"
                                      },
                                      true
                                   ]
                                }
                             ],
                             "type": "tree",
                             "rules": [
                                {
                                   "conditions": [
                                      {
                                         "fn": "booleanEquals",
                                         "argv": [
                                            true,

                                            {
                                               "fn": "getAttr",
                                               "argv": [
                                                  {
                                                     "ref": "PartitionResult"
                                                  },
                                                  "supportsDualStack"
                                               ]
                                            }
                                         ]
                                      }
                                   ],

           "type": "tree",
                                   "rules": [
                                      {
                                         "conditions": [],
                                         "type": "tree",
                                         "rules": [
```

```
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},

                    "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ]
                            }
                        ]
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }

                ]
                },
                {
                    "conditions": [],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ]
                }
            ]
        }
    ]
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
```

```
                    ]
                }
            ]
        },
        "smithy.rules#endpointTests": {
            "testCases": [
                {
                    "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                        "endpoint": {

             "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                        }
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseDualStack": true,
                        "UseFIPS": true
                    }
                },
                {
                    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                        }
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseDualStack": false,
                        "UseFIPS":
true
                    }
                },
                {
                    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                        }
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseDualStack": true,
                        "UseFIPS": false
                    }
                },
                {
```

```
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseDualStack": false,
          "UseFIPS": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseDualStack": true,
          "UseFIPS": true

        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseDualStack": false,
          "UseFIPS": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
```

```
          "Region": "cn-north-1",
          "UseDualStack": true,
          "UseFIPS": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseDualStack": false,
          "UseFIPS": false


}
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseDualStack": true,
          "UseFIPS": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseDualStack": false,
          "UseFIPS": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
```

          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseDualStack": true,
          "UseFIPS": false

}
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseDualStack": false,
          "UseFIPS": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseDualStack": false,
          "UseFIPS": true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseDualStack": false,
          "UseFIPS": false

```
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
          }
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseDualStack": false,
          "UseFIPS": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
          }
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseDualStack": false,
          "UseFIPS": false
        }
      },
      {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
          "endpoint": {
            "url": "https://example.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseDualStack": false,
          "UseFIPS": false,
          "Endpoint":
"https://example.com"
        }
      },
      {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
```

```
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseDualStack": false,
              "UseFIPS": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseDualStack": false,
              "UseFIPS": true,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseDualStack": true,
              "UseFIPS": false,
              "Endpoint": "https://example.com"
            }
          }

        ],
        "version": "1.0"
      }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
```

```
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/ListIdentityProviders"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize":
"MaxResults",
    "items": "IdentityProviderSummaries"
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
      }
    },
```

```
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
      "type": "structure",
      "members": {

        "IdentityProviderSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
            "smithy.api#required": {}
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
```

```
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number
of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify.</p>"
          }
        }
```

```
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
            "type": "structure",
            "members": {
              "InstanceSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
                }
              },
              "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the user-based
   subscription products available from an identity provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "ProductUserSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },

            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
   that match one or\n      more sets
```

of key-value pairs that you specify.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
            "type": "structure",
            "members": {
                "ProductUserSummaries": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                    "traits": {
                        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation":
"<p>Token for the next set of results.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },

```
        {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize":
"MaxResults",
          "items": "InstanceUserSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
```

```
                  "MaxResults":
{
                      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                      "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                      }
                  },
                  "Filters": {
                      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                      "traits": {
                        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify.</p>"
                      }
                  },
                  "NextToken": {
                      "target": "smithy.api#String",
                      "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                      }
                  }
              }
          },
          "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse":
{
              "type": "structure",
              "members": {
                  "InstanceUserSummaries": {
                      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                      "traits": {
                        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                      }
                  },
                  "NextToken": {
                      "target": "smithy.api#String",
                      "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                      }
                  }
              }
          },
          "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
              "type": "structure",
              "members": {
                  "Username": {
                      "target": "smithy.api#String",
                      "traits": {
                        "smithy.api#documentation":
```

"<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {

  "smithy.api#documentation": "<p>The status of a product for a user.</p>",
                "smithy.api#required": {}
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            },
            "SubscriptionStartDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
              }
            },
            "SubscriptionEndDate": {
              "target":
"smithy.api#String",
              "traits": {

```
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
```

```
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product":
  {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n     subnets to provision VPC endpoints.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",

      "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
```

```
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      },

      "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
```

```
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n     your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n     subnets to provision VPC endpoints, and the security group ID that is associated with the VPC\n
endpoints. The security group should permit inbound TCP port 1688 communication from resources\n
 in the VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
           },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n      <note>\n        <p>Your estimated bill for charges on the number of users and related costs
will take 48\n      hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
  monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n      </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An
object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the user.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
```

```
          "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
           "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
           }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
         {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
        "smithy.api#http": {
          "code": 200,
```

```
            "method": "POST",
            "uri":
"/user/StopProductSubscription"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
```

```
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>The request was denied because of request throttling. Retry the request.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity\n     provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/UpdateIdentityProviderSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#required": {}


      }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "UpdateSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
        "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can\n     update any combination of settings in a single operation such as the:</p>\n <ul>\n     <li>\n          <p>Subnets which you want to add to provision VPC endpoints.</p>\n        </li>\n     <li>\n          <p>Subnets which you want to remove the VPC endpoints from.</p>\n        </li>\n <li>\n

        <p>Security group ID which permits traffic to the VPC endpoints.</p>\n        </li>\n     </ul>",
          "smithy.api#required": {}
        }
```

```
                }
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
              "type": "structure",
              "members": {
                "IdentityProviderSummary": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                  "traits": {
                    "smithy.api#required": {}
                  }
                }
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
              "type": "structure",
              "members": {
                "AddSubnets": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                  "traits": {
                    "smithy.api#documentation":
 "<p>The ID of one or more subnets in which License Manager will create a VPC endpoint for products that\n
require connectivity to activation servers.</p>",
                    "smithy.api#required": {}
                  }
                },
                "RemoveSubnets": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                  "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                  }
                },
                "SecurityGroupId": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                  "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n      your VPC and the VPC endpoints for activation servers.</p>"

                  }
                }
              },
              "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as\n      the subnets to provision VPC endpoints.</p>"
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
```

```
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
          }
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
```

```go
    // This member is required.
    LicenseArn *string

    // Current version of the
    license.
    //
    // This member is required.
    SourceVersion *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack,
    options); err != nil {
        return err
    }
```

```
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
 if err = addHTTPSignerV4Middleware(stack, options); err != nil {
 return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addClientUserAgent(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack);
 err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  SigningName:   "license-manager",
  OperationName: "DeleteLicense",
 }
}
```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

                        Apache License
                  Version 2.0, January 2004
                http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

```
{
  "smithy": "2.0",
  "metadata": {
    "suppressions": [
      {
        "id": "HttpMethodSemantics",
        "namespace": "*"
      },
      {
        "id": "HttpResponseCodeSemantics",
        "namespace": "*"
      },
      {
        "id": "PaginatedTrait",
        "namespace": "*"
      },
      {
        "id": "HttpHeaderTrait",
        "namespace": "*"
      },
      {
        "id": "HttpUriConflict",
        "namespace": "*"
      },
      {
        "id": "Service",
        "namespace": "*"
      }
    ]
  },
  "shapes": {
    "com.amazonaws.licensemanager#AWSLicenseManager": {
      "type": "service",
      "version": "2018-08-01",
      "operations": [
        {
          "target": "com.amazonaws.licensemanager#AcceptGrant"
        },

        {
          "target": "com.amazonaws.licensemanager#CheckInLicense"
        },
        {
```

```
    "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
  },
  {
    "target": "com.amazonaws.licensemanager#CheckoutLicense"
  },
  {
    "target": "com.amazonaws.licensemanager#CreateGrant"
  },
  {
    "target": "com.amazonaws.licensemanager#CreateGrantVersion"
  },
  {
    "target": "com.amazonaws.licensemanager#CreateLicense"
  },
  {
    "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
  },
  {
    "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
  },
  {
    "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
  },
  {
    "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
  },
  {
    "target": "com.amazonaws.licensemanager#CreateToken"
  },
  {
    "target": "com.amazonaws.licensemanager#DeleteGrant"
  },
  {
    "target": "com.amazonaws.licensemanager#DeleteLicense"
  },
  {
    "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
  },
  {
    "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
  },
  {
    "target": "com.amazonaws.licensemanager#DeleteToken"
  },
  {
    "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
```

```
        },
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
    {
      "target": "com.amazonaws.licensemanager#GetGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseUsage"
    },
    {
      "target": "com.amazonaws.licensemanager#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
     },
    {
      "target": "com.amazonaws.licensemanager#ListDistributedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
```

    },
    {

      "target": "com.amazonaws.licensemanager#ListLicenseVersions"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListResourceInventory"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTokens"
    },
    {
      "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#RejectGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
    },
    {

```
                    "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
                }
            ],
            "traits": {
                "aws.api#service": {
                    "sdkId": "License Manager",
                    "arnNamespace": "license-manager",
                    "cloudFormationName": "LicenseManager",
                    "cloudTrailEventSource": "licensemanager.amazonaws.com",
                    "endpointPrefix": "license-manager"
                },
                "aws.auth#sigv4": {
                    "name": "license-manager"
                },
                "aws.protocols#awsJson1_1": {},
                "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n        Amazon Web Services accounts and on-premises servers.</p>",
                "smithy.api#title": "AWS License Manager",
                "smithy.api#xmlNamespace": {
                    "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
                },
                "smithy.rules#endpointRuleSet": {
                    "version": "1.0",
                    "parameters": {
                        "Region": {
                            "builtIn": "AWS::Region",
                            "required": false,
                            "documentation": "The AWS region used to dispatch the request.",
                            "type": "String"
                        },
                        "UseDualStack": {
                            "builtIn": "AWS::UseDualStack",
                            "required": true,
                            "default": false,
                            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                            "type": "Boolean"
                        },
                        "UseFIPS": {
                            "builtIn": "AWS::UseFIPS",
                            "required": true,
                            "default": false,
                            "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
                            "type": "Boolean"
                        },
                        "Endpoint": {
```

```
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]

                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                },
                {

"conditions": [],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
```

```
                },
                  true
                ]
              }
            ],
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",

  "type": "error"
          },
          {
            "conditions": [],
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },
              "properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ]
      }
    ]
  },
  {
    "conditions": [],
    "type": "tree",
    "rules": [
      {

        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                "argv": [
                  {
```

```
          "ref": "Region"
                              }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
          true
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
```

                    "supportsFIPS"
                                ]
                              }
                            ]
                          },
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              true,
                              {
                                "fn": "getAttr",
                                "argv": [
                                  {

                    "ref": "PartitionResult"
                                  },
                                  "supportsDualStack"
                                ]
                              }
                            ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
                          {
                            "conditions": [],
                            "type": "tree",
                            "rules": [
                              {
                                "conditions": [],
                                "endpoint": {
                                  "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
                                "type": "endpoint"
                              }

        ]
                          }
                        ]
                      },
                      {
                        "conditions": [],
                        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                        "type": "error"

```json
                              }
                          ]
                      },
                      {
                          "conditions": [
                              {
                                  "fn": "booleanEquals",

                                  "argv": [
                                      {
                                          "ref": "UseFIPS"
                                      },
                                      true
                                  ]
                              }
                          ],
                          "type": "tree",
                          "rules": [
                              {
                                  "conditions": [
                                      {
                                          "fn": "booleanEquals",

            "argv": [

                                              true,
                                              {
                                                  "fn": "getAttr",
                                                  "argv": [
                                                      {
                                                          "ref": "PartitionResult"
                                                      },
                                                      "supportsFIPS"
                                                  ]
                                              }

    }

                                  ],
                                  "type": "tree",
                                  "rules": [
                                      {
                                          "conditions": [],
                                          "type": "tree",
                                          "rules": [
                                              {
                                                  "conditions": [],
                                                  "endpoint": {
                                                      "url": "https://license-manager-
```

fips.{Region}.{PartitionResult#dnsSuffix}",

                    "properties": {},
                                        "headers": {}
                                    },
                                  "type": "endpoint"
                              }
                          ]
                      }
                  ]
              },
              {
                  "conditions": [],
                  "error": "FIPS is enabled but this partition does not support FIPS",
                  "type": "error"

          }
          ]
      },
      {
          "conditions": [
              {
                  "fn": "booleanEquals",
                  "argv": [
                      {
                          "ref": "UseDualStack"
                      },
                      true
                  ]
              }
          ],
          "type":

"tree",

          "rules": [
              {
                  "conditions": [
                      {
                          "fn": "booleanEquals",
                          "argv": [
                              true,
                              {
                                  "fn": "getAttr",
                                  "argv": [
                                      {
                                          "ref": "PartitionResult"

              },

                      "supportsDualStack"

```
                                    ]
                                  }
                                ]
                              }
                            ],
                            "type": "tree",
                            "rules": [
                              {
                                "conditions": [],
                                "type": "tree",
                                "rules": [

                  {
                                      "conditions": [],
                                      "endpoint": {
                                        "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                          "properties": {},
                                          "headers": {}
                                      },
                                      "type": "endpoint"
                                  }
                                ]
                              }

          ]
                        },
                        {
                          "conditions": [],
                          "error": "DualStack is enabled but this partition does not support DualStack",
                          "type": "error"
                        }
                      ]
                    },
                    {
                      "conditions": [],
                      "type": "tree",
                      "rules": [
                        {
                          "conditions": [],

                  "endpoint": {
                            "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                          },
                          "type": "endpoint"
                        }
```

```
                                  ]
                                }
                              ]
                            }
                          ]
                        },
                        {
                          "conditions": [],
                          "error": "Invalid Configuration: Missing Region",

              "type": "error"
                        }
                      ]
                  }
                ]
              },
            "smithy.rules#endpointTests": {
              "testCases": [
                {
                  "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager.af-south-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region": "af-south-1"
                  }
                },
                {
                  "documentation": "For region ap-east-1 with FIPS disabled
and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager.ap-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region": "ap-east-1"
                  }
                },
                {
                  "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
                  "expect": {
```

```
              "endpoint": {
                "url": "https://license-manager.ap-northeast-1.amazonaws.com"
              }
            },
            "params": {
              "UseFIPS":

false,

              "UseDualStack": false,
              "Region": "ap-northeast-1"
            }
          },
          {
            "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.ap-northeast-2.amazonaws.com"
              }
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Region": "ap-northeast-2"
            }
          },
          {
            "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",

    "expect": {

              "endpoint": {
                "url": "https://license-manager.ap-northeast-3.amazonaws.com"
              }
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Region": "ap-northeast-3"
            }
          },
          {
            "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.ap-south-1.amazonaws.com"
              }
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack":
```

false,
                    "Region": "ap-south-1"
                  }
                },
                {
                  "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager.ap-southeast-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region": "ap-southeast-1"
                  }
                },
                {
                  "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint":
{
                      "url": "https://license-manager.ap-southeast-2.amazonaws.com"
                    }
                  },
                  "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region": "ap-southeast-2"
                  }
                },
                {
                  "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager.ap-southeast-3.amazonaws.com"
                    }
                  },
                  "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region":
"ap-southeast-3"
                  }
                },
                {
                  "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
                  "expect": {

```
        "endpoint": {
          "url": "https://license-manager.ca-central-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "ca-central-1"
      }
    },
    {
      "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.eu-central-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "eu-central-1"
      }
    },
    {
      "documentation": "For region eu-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.eu-north-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "eu-north-1"
      }

},
    {
      "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.eu-south-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "eu-south-1"
```

```
                }
            },
            {
                "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.eu-west-1.amazonaws.com"

    }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Region": "eu-west-1"
            }
        },
        {
            "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.eu-west-2.amazonaws.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Region": "eu-west-2"
            }
        },
        {
            "documentation": "For
region eu-west-3 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.eu-west-3.amazonaws.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Region": "eu-west-3"
            }
        },
        {
            "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.me-south-1.amazonaws.com"
```

```
                }
              },
              "params": {

        "UseFIPS": false,
                "UseDualStack": false,
                "Region": "me-south-1"
              }
            },
            {
              "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.sa-east-1.amazonaws.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Region": "sa-east-1"
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",

      "expect": {
                "endpoint": {
                  "url": "https://license-manager.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Region": "us-east-1"
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "UseFIPS": true,
                "UseDualStack":
false,
                "Region": "us-east-1"
```

```
                }
            },
            {
                "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-east-2.amazonaws.com"
                    }
                },
                "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region": "us-east-2"
                }
            },
            {
                "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {

        "url": "https://license-manager-fips.us-east-2.amazonaws.com"
                    }
                },
                "params": {
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Region": "us-east-2"
                }
            },
            {
                "documentation": "For region us-west-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-west-1.amazonaws.com"
                    }
                },
                "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region": "us-west-1"

}
            },
            {
                "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-west-1.amazonaws.com"
```

```
                    }
                },
                "params": {
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Region": "us-west-1"
                }
            },
            {
                "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-west-2.amazonaws.com"

                }
                },
                "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region": "us-west-2"
                }
            },
            {
                "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-west-2.amazonaws.com"
                    }
                },
                "params": {
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Region": "us-west-2"
                }
            },
            {

"documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-east-1.api.aws"
                    }
                },
                "params": {
                    "UseFIPS": true,
                    "UseDualStack": true,
                    "Region": "us-east-1"
                }
```

```
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-east-1.api.aws"
        }
      },
      "params": {

        "UseFIPS": false,
        "UseDualStack": true,
        "Region": "us-east-1"
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "cn-north-1"
      }
    },
    {
      "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",

      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "cn-northwest-1"
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
```

```
        },
        "params": {
          "UseFIPS": true,
           "UseDualStack": true,
          "Region": "cn-north-1"
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
             "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "UseFIPS": true,
          "UseDualStack": false,
          "Region": "cn-north-1"
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {

"endpoint": {
              "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "UseFIPS": false,
          "UseDualStack": true,
          "Region": "cn-north-1"
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
             "url": "https://license-manager.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": false,
          "UseDualStack": false,


          "Region": "us-gov-east-1"
        }
      },
```

```
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "UseFIPS": true,
        "UseDualStack": false,
        "Region": "us-gov-east-1"
    }
},
{
    "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {

            "url": "https://license-manager.us-gov-west-1.amazonaws.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "us-gov-west-1"
    }
},
{
    "documentation": "For region us-gov-west-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
        }
    },
    "params": {
        "UseFIPS": true,
        "UseDualStack": false,
        "Region": "us-gov-west-1"
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
```

```
          "UseFIPS": true,
          "UseDualStack": true,
          "Region": "us-gov-east-1"
        }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
           "url": "https://license-manager.us-gov-east-1.api.aws"
         }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": true,
        "Region": "us-gov-east-1"
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
           "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
         }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "us-iso-east-1"
      }
    },

    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
      },
      "params": {
        "UseFIPS": true,
        "UseDualStack": true,
        "Region": "us-iso-east-1"
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
           "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
```

```
              }
          },

"params": {
            "UseFIPS": true,
            "UseDualStack": false,
            "Region": "us-iso-east-1"
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": true,
            "Region": "us-iso-east-1"
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Region": "us-isob-east-1"
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
            "UseFIPS": true,
            "UseDualStack": true,
            "Region": "us-isob-east-1"
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
```

              "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "UseFIPS": true,
            "UseDualStack": false,
            "Region": "us-isob-east-1"
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": true,
            "Region": "us-isob-east-1"
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Region": "us-east-1",
            "Endpoint": "https://example.com"
          }
        },

        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }

          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {

        "UseFIPS": true,
              "UseDualStack": false,
              "Region": "us-east-1",
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack": true,
              "Region": "us-east-1",
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "Missing region",

           "expect": {
              "error": "Invalid Configuration: Missing Region"
            }
          }
        ],
          "version": "1.0"
      }
    }
  },
  "com.amazonaws.licensemanager#AcceptGrant": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#AcceptGrantRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
    },
    "errors": [
      {

```
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },

  {

            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",

"smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant version.</p>"

        }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#AccessDeniedException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "ServiceAccessDenied",
                "httpResponseCode": 401
            },
            "smithy.api#documentation": "<p>Access to resource denied.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 401
        }
    },
    "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
        "type": "enum",
        "members": {
            "DISTRIBUTED_GRANTS_ONLY": {
                "target": "smithy.api#Unit",

"traits": {
                    "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
                }
            },
```

```
        "ALL_GRANTS_PERMITTED_BY_ISSUER": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#AllowedOperation": {
      "type": "enum",
      "members": {
        "CREATE_GRANT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateGrant"
          }
        },
        "CHECKOUT_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutLicense"
          }
        },
        "CHECKOUT_BORROW_LICENSE":
{
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutBorrowLicense"
          }
        },
        "CHECK_IN_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckInLicense"
          }
        },
        "EXTEND_CONSUMPTION_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ExtendConsumptionLicense"
          }
        },
        "LIST_PURCHASED_LICENSES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ListPurchasedLicenses"
          }
        },
```

```
        "CREATE_TOKEN": {
          "target": "smithy.api#Unit",


         "traits": {
             "smithy.api#enumValue": "CreateToken"
           }
         }
       }
      },
     "com.amazonaws.licensemanager#AllowedOperationList": {
       "type": "list",
       "member": {
         "target": "com.amazonaws.licensemanager#AllowedOperation"
       },
       "traits": {
         "smithy.api#length": {
           "min": 1,
           "max": 7
         }
       }
     },
     "com.amazonaws.licensemanager#Arn": {
       "type": "string",
       "traits": {
         "smithy.api#length": {
           "min": 0,
           "max": 2048
         },
         "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
       }
     },
     "com.amazonaws.licensemanager#ArnList":
{
       "type": "list",
       "member": {
         "target": "com.amazonaws.licensemanager#Arn"
       }
     },
     "com.amazonaws.licensemanager#AuthorizationException": {
       "type": "structure",
       "members": {
         "Message": {
           "target": "com.amazonaws.licensemanager#Message"
         }
       },
       "traits": {
         "aws.protocols#awsQueryError": {
```

```
                  "code": "AuthorizationFailure",
                  "httpResponseCode": 403
              },
              "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n          policy associated with this account.</p>",
              "smithy.api#error": "client",
              "smithy.api#httpError": 403
          }
      },
      "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
          "type":
"structure",
          "members": {
              "LastRunTime": {
                  "target": "com.amazonaws.licensemanager#DateTime",
                  "traits": {
                      "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
                  }
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>Describes automated discovery.</p>"
          }
      },
      "com.amazonaws.licensemanager#Boolean": {
          "type": "boolean",
          "traits": {
              "smithy.api#default": false
          }
      },
      "com.amazonaws.licensemanager#BorrowConfiguration": {
          "type": "structure",
          "members": {
              "AllowEarlyCheckIn": {
                  "target": "com.amazonaws.licensemanager#BoxBoolean",
                  "traits": {
                      "smithy.api#documentation": "<p>Indicates whether early check-ins are
allowed.</p>",
                      "smithy.api#required": {}
                  }
              },
              "MaxTimeToLiveInMinutes": {
                  "target": "com.amazonaws.licensemanager#BoxInteger",
                  "traits": {
                      "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
                      "smithy.api#required": {}
                  }
              }
```

```
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#BoxBoolean": {
      "type": "boolean"
    },
    "com.amazonaws.licensemanager#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanager#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#CheckInLicense": {
      "type": "operation",
      "input":
{
        "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {

          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
          }
        },
        "com.amazonaws.licensemanager#CheckInLicenseRequest": {
          "type": "structure",
          "members": {
            "LicenseConsumptionToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License consumption token.</p>",
                "smithy.api#required": {}
              }
            },
            "Beneficiary": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {

                "smithy.api#documentation": "<p>License beneficiary.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#CheckInLicenseResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#CheckoutBorrowLicense": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
```

```
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
         },
        {
          "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
         },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
```

```
                "traits":
{
                    "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
                    "smithy.api#required": {}
                }
            },
            "DigitalSignatureMethod": {
                "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
                "traits": {
                    "smithy.api#documentation": "<p>Digital signature method. The possible value is JSON Web
Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
                    "smithy.api#required": {}
                }
            },
            "NodeId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Node ID.</p>"
                }

            },
            "CheckoutMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Information about constraints.</p>"
                }
            },
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
        "type": "structure",
        "members": {
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
```

```
                }
            },
            "LicenseConsumptionToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License consumption token.</p>"
                }
            },
            "EntitlementsAllowed": {
                "target": "com.amazonaws.licensemanager#EntitlementDataList",
                "traits": {
                    "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
                }
            },
            "NodeId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Node ID.</p>"
                }
            },
            "SignedToken":
{

                "target": "com.amazonaws.licensemanager#SignedToken",
                "traits": {
                    "smithy.api#documentation": "<p>Signed token.</p>"
                }
            },
            "IssuedAt": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
                }
            },
            "Expiration": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
                }
            },
            "CheckoutMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation":
"<p>Information about constraints.</p>"
                }
            }
        },
        "traits": {
```

```
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n          <p>If the account that created the license
 is the same that is performing the check out, you must\n          specify the account as the beneficiary.</p>\n
</note>"
```

```
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
        "type": "structure",
        "members": {
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product SKU.</p>",
              "smithy.api#required": {}
            }
          },
          "CheckoutType": {
            "target": "com.amazonaws.licensemanager#CheckoutType",
            "traits": {
              "smithy.api#documentation": "<p>Checkout type.</p>",
              "smithy.api#required": {}
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",

            "traits": {
              "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>",
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
             "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          },
```

```
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
},
"com.amazonaws.licensemanager#CheckoutLicenseResponse": {
  "type": "structure",
  "members": {
    "CheckoutType": {
      "target": "com.amazonaws.licensemanager#CheckoutType",
      "traits": {
        "smithy.api#documentation": "<p>Checkout type.</p>"
      }
    },
    "LicenseConsumptionToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>License consumption token.</p>"
      }
    },
    "EntitlementsAllowed": {
      "target": "com.amazonaws.licensemanager#EntitlementDataList",
      "traits": {
        "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
      }
    },
    "SignedToken": {
      "target": "com.amazonaws.licensemanager#SignedToken",
      "traits": {
        "smithy.api#documentation": "<p>Signed token.</p>"
      }
    },
    "NodeId": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Node ID.</p>"
      }
    },
    "IssuedAt": {
      "target": "com.amazonaws.licensemanager#ISO8601DateTime",
      "traits": {
        "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
```

```
                }
              },
              "Expiration": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                  "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
                }
              },
              "LicenseArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
                }
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#CheckoutType":
{
          "type": "enum",
          "members": {
            "PROVISIONAL": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "PROVISIONAL"
              }
            },
            "PERPETUAL": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "PERPETUAL"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#ClientRequestToken": {
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 36
            }
          }
        },
        "com.amazonaws.licensemanager#ClientToken": {
          "type": "string",
```

```
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 2048
          },
          "smithy.api#pattern": "^\\S+$"
        }
      },
      "com.amazonaws.licensemanager#ConflictException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "ConflictException",
            "httpResponseCode": 409
          },
          "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 409
        }
      },
      "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
        "type": "structure",
        "members": {
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits":
{
              "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
            }
          },
          "ConsumedLicenses": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about license consumption.</p>"
        }
      },
      "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
```

```
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
      }
    },
    "com.amazonaws.licensemanager#ConsumptionConfiguration": {
      "type": "structure",
      "members": {
        "RenewType": {

  "target": "com.amazonaws.licensemanager#RenewType",
          "traits": {
            "smithy.api#documentation": "<p>Renewal frequency.</p>"
          }
        },
        "ProvisionalConfiguration": {
          "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
          }
        },
        "BorrowConfiguration": {
          "target": "com.amazonaws.licensemanager#BorrowConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrant":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
```

```
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
             },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more information, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/granted-licenses.html\">Granted licenses in License Manager</a> in the <i>License
Manager User Guide</i>.</p>"
          }
        },
        "com.amazonaws.licensemanager#CreateGrantRequest": {
          "type": "structure",
          "members": {
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientToken",
              "traits": {
                "smithy.api#documentation":
 "<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            },
            "GrantName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant name.</p>",
                "smithy.api#required": {}
              }
            },
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                "smithy.api#required": {}
              }
```

```
        },
        "Principals": {
          "target": "com.amazonaws.licensemanager#PrincipalArnList",
          "traits": {
            "smithy.api#documentation":
  "<p>The grant principals. You can specify one of the following as an Amazon Resource Name\n
(ARN):</p>\n       <ul>\n         <li>\n            <p>An Amazon Web Services account, which includes only the
account specified.</p>\n         </li>\n      </ul>\n      <ul>\n         <li>\n            <p>An organizational unit
(OU), which includes all accounts in the OU.</p>\n         </li>\n      </ul>\n      <ul>\n         <li>\n
<p>An organization, which will include all accounts across your organization.</p>\n         </li>\n      </ul>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
            "smithy.api#required": {}

    }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation":
  "<p>Grant status.</p>"
          }
```

```
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateGrantVersion": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
        "type": "structure",
        "members": {
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          },
          "GrantName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

              "smithy.api#documentation": "<p>Grant name.</p>"
            }
          },
          "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>Grant status reason.</p>"
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Current version of the grant.</p>"
          }
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
            "smithy.api#documentation": "<p>The options specified for the grant.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>New version of the grant.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
```

```
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"

      },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
```

```
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"

        }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or all of these), license affinity  to
host (how long a \n        license must be associated with a host), and the number of licenses purchased and
used.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Name of the license configuration.</p>",
                    "smithy.api#required": {}
                }
            },
            "Description": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Description of the license configuration.</p>"
                }
            },
            "LicenseCountingType": {
                "target": "com.amazonaws.licensemanager#LicenseCountingType",
                "traits": {
                    "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseCount": {
                "target": "com.amazonaws.licensemanager#BoxLong",
                "traits": {
                    "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
```

```
            }
          },
          "LicenseCountHardLimit": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n        blocks the launch of new instances.</p>"
            }
          },
          "LicenseRules": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n        vary by dimension, as follows.</p>\n
<ul>\n        <li>\n          <p>\n            <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
      <code>licenseAffinityToHost</code> |\n          <code>maximumCores</code> |
<code>minimumCores</code>\n
          </p>\n        </li>\n        <li>\n          <p>\n            <code>Instances</code> dimension:
<code>allowedTenancy</code> | \n          <code>maximumCores</code> | <code>minimumCores</code> |\n
      <code>maximumSockets</code> | <code>minimumSockets</code> |\n
<code>maximumVcpus</code> | <code>minimumVcpus</code>\n          </p>\n        </li>\n        <li>\n
    <p>\n            <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n          <code>maximumSockets</code> |
<code>minimumSockets</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n          <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n          </p>\n
  </li>\n      </ul>\n      <p>The unit for <code>licenseAffinityToHost</code> is days and the range is 1 to 180.
The possible \n      values for <code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-
DedicatedHost</code>, and \n      <code>EC2-DedicatedInstance</code>. The possible values for
<code>honorVcpuOptimization</code> are \n      <code>True</code> and <code>False</code>.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
            }
          },
          "DisassociateWhenNotFound": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
            }
          },
          "ProductInformationList": {
            "target": "com.amazonaws.licensemanager#ProductInformationList",
```

```
              "traits": {
                 "smithy.api#documentation": "<p>Product information.</p>"
              }
           }
        },
        "traits": {
           "smithy.api#input": {}
        }
     },
     "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse": {
        "type": "structure",
        "members": {
           "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
              }
           }
        },
        "traits": {
           "smithy.api#output": {}
        }
     },
     "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource":
{
        "type": "operation",
        "input": {
           "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
        },
        "output": {
           "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
        },
        "errors": [
           {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
           },
           {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
           },
           {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
           },
           {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
           },
           {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation":
 "<p>Information that identifies the license type you are converting from. \n\n        For the structure of the source
license, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/conversion-
procedures.html#conversion-cli\">Convert a license type using the CLI </a> in the <i>License Manager User
Guide</i>.</p>",
              "smithy.api#required": {}
            }
          },
          "DestinationLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",

              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
```

```
    "type": "structure",
    "members": {
      "LicenseConversionTaskId": {
        "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
```

```
          "smithy.api#documentation": "<p>Creates a report generator.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "ReportGeneratorName": {
            "target": "com.amazonaws.licensemanager#ReportGeneratorName",
            "traits": {
              "smithy.api#documentation": "<p>Name of the report generator.</p>",
              "smithy.api#required": {}
            }
          },
          "Type": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
              "smithy.api#documentation": "<p>Type of reports to generate. The
 following report types an be generated:</p>\n        <ul>\n          <li>\n            <p>License configuration report -
Reports the number and details of consumed licenses for a license configuration.</p>\n          </li>\n          <li>\n
      <p>Resource report - Reports the tracked licenses and resource consumption for a license
configuration.</p>\n          </li>\n        </ul>",
              "smithy.api#required": {}
            }
          },
          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",

    "traits": {
              "smithy.api#documentation": "<p>Frequency by which reports are generated. Reports can be
generated daily, monthly, or weekly.</p>",
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientRequestToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
```

            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanager#TagList",
               "traits": {
                "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
          "type": "structure",
          "members": {
            "LicenseManagerReportGeneratorArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseRequest": {
          "type": "structure",
          "members": {

  "LicenseName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License name.</p>",
                "smithy.api#required": {}
              }
            },
            "ProductName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Product name.</p>",

```
              "smithy.api#required": {}
          }
      },
      "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
              "smithy.api#documentation": "<p>Product SKU.</p>",
              "smithy.api#required": {}
          }
      },
      "Issuer": {
          "target": "com.amazonaws.licensemanager#Issuer",

  "traits": {
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
          }
      },
      "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
              "smithy.api#documentation": "<p>Home Region for the license.</p>",
              "smithy.api#required": {}
          }
      },
      "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
              "smithy.api#required": {}
          }
      },
      "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits":
{
              "smithy.api#documentation": "<p>License entitlements.</p>",
              "smithy.api#required": {}
          }
      },
      "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>",
              "smithy.api#required": {}
          }
      },
```

```
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
            "smithy.api#required": {}


          }
        },
        "LicenseMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
```

```
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#CreateLicenseVersion": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#RedirectException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
  }
},
"com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
  "type": "structure",
```

```json
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required":
{}
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product name.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits": {
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Validity": {

          "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-UTC format.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Information about the license.</p>"
            }
        },
        "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
                "smithy.api#documentation": "<p>License entitlements.</p>",
                "smithy.api#required": {}
            }
        },
        "ConsumptionConfiguration": {
            "target":
"com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
                "smithy.api#documentation": "<p>License status.</p>",
                "smithy.api#required": {}
            }
        },
        "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you
provide to ensure the idempotency of the request.</p>",
                "smithy.api#required": {}
            }
        },
        "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Current version of the license.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
```

```
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",


            "traits": {
              "smithy.api#documentation": "<p>New version of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateToken": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {

          "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
```

```
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get
 an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get role credentials that
you can use to \n        call License Manager to manage the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateTokenRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
            "smithy.api#required": {}
          }
        },
        "RoleArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n
    License Manager does not check whether the roles are in use.</p>"
          }
        },
        "ExpirationInDays": {
          "target": "com.amazonaws.licensemanager#Integer",
          "traits": {
            "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
```

```json
          "traits": {
            "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The data is mapped\n          to the amr claim of the JWT token.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateTokenResponse": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
        "TokenType": {
          "target": "com.amazonaws.licensemanager#TokenType",
          "traits": {
            "smithy.api#documentation": "<p>Token type.</p>"
          }
        },
        "Token": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {

            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DateTime": {
      "type": "timestamp"
    },
```

```
"com.amazonaws.licensemanager#DatetimeRange": {
    "type": "structure",
    "members": {
        "Begin": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>Start of the time range.</p>",
                "smithy.api#required": {}
            }
        },
        "End": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>End of the time range.</p>"

}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
    }
},
"com.amazonaws.licensemanager#DeleteGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes the specified grant.</p>"
    }
  },
  "com.amazonaws.licensemanager#DeleteGrantRequest": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "StatusReason":
{
        "target": "com.amazonaws.licensemanager#StatusReasonMessage",
        "traits": {
          "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
        }
      },
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Current version of the grant.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#DeleteGrantResponse": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Grant ARN.</p>"
        }
      },
      "Status": {
```

```
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
            }
        },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#DeleteLicense": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
    },
    "errors": [

        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
```

```
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {

        "smithy.api#documentation": "<p>Deletes the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },

  {

          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the license configuration.</p>",
            "smithy.api#required": {}
          }
```

```
                }
              },
              "traits": {
                "smithy.api#input": {}
              }
            },
            "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {

"type": "structure",
              "members": {},
              "traits": {
                "smithy.api#output": {}
              }
            },
            "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
              "type": "operation",
              "input": {
                "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
              },
              "output": {
                "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
              },
              "errors": [
                {
                  "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                  "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                  "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                  "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                 },
                {
                  "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                  "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                  "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                  "target": "com.amazonaws.licensemanager#ValidationException"
                }
              ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {

    "type": "structure",
        "members": {
            "LicenseManagerReportGeneratorArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseRequest": {
        "type": "structure",
        "members": {
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits":
{
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                    "smithy.api#required": {}
                }
            },
            "SourceVersion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Current version of the license.</p>",
                    "smithy.api#required": {}
                }
            }
```

```
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseResponse": {
      "type": "structure",
      "members": {
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }

        },
          "DeletionDate": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteToken": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {

          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
```

```
                  {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                  },
                  {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                  },
                  {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                  }
                ],
                "traits": {
                  "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
                }
              },
              "com.amazonaws.licensemanager#DeleteTokenRequest": {
                "type": "structure",
                "members": {
                  "TokenId": {
                    "target": "com.amazonaws.licensemanager#String",

                  "traits": {
                        "smithy.api#documentation": "<p>Token ID.</p>",
                        "smithy.api#required": {}
                    }
                  }
                },
                "traits": {
                  "smithy.api#input": {}
                }
              },
              "com.amazonaws.licensemanager#DeleteTokenResponse": {
                "type": "structure",
                "members": {},
                "traits": {
                  "smithy.api#output": {}
                }
              },
              "com.amazonaws.licensemanager#DigitalSignatureMethod": {
                "type": "enum",
                "members": {
                  "JWT_PS384": {
                    "target": "smithy.api#Unit",
                    "traits": {
                      "smithy.api#enumValue": "JWT_PS384"
                    }
                  }
                }
```

        },
      "com.amazonaws.licensemanager#Entitlement": {
        "type": "structure",
        "members": {
          "Name": {

        "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Entitlement name.</p>",
                "smithy.api#required": {}
              }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement resource. Use only if the unit is None.</p>"
            }
          },
          "MaxCount": {
            "target": "com.amazonaws.licensemanager#Long",
            "traits": {
              "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
            }
          },
          "Overage": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation":
  "<p>Indicates whether overages are allowed.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement unit.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowCheckIn": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
        }

```
    },
    "com.amazonaws.licensemanager#EntitlementData": {
      "type": "structure",
      "members": {
        "Name":

{

          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement data name.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement data value.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement data unit.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"

}
    },
    "com.amazonaws.licensemanager#EntitlementDataList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementData"
      }
    },
    "com.amazonaws.licensemanager#EntitlementDataUnit": {
      "type": "enum",
      "members": {
        "COUNT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count"
          }
        },
        "NONE": {
          "target": "smithy.api#Unit",
```

```json
      "traits": {
        "smithy.api#enumValue": "None"
      }
    },
    "SECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Seconds"
      }
    },
    "MICROSECONDS": {

      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Microseconds"
      }
    },
    "MILLISECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Milliseconds"
      }
    },
    "BYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"

}
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
```

```
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits"
      }
    },
    "KILOBITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobits"
      }
    },
    "MEGABITS": {
      "target": "smithy.api#Unit",
      "traits": {

"smithy.api#enumValue": "Megabits"
      }
    },
    "GIGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabits"
      }
    },
    "TERABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabits"
      }
    },
    "PERCENT": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Percent"
      }
    },
    "BYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes/Second"
      }
```

```
        },
        "KILOBYTES_PER_SECOND": {
          "target":
"smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabytes/Second"
          }
        },
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
```

```
              "smithy.api#enumValue": "Gigabits/Second"
            }
          },
          "TERABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Terabits/Second"
            }
          },

      "COUNT_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count/Second"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#EntitlementList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Entitlement"
        }
      },
      "com.amazonaws.licensemanager#EntitlementNotAllowedException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#EntitlementUnit": {

  "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
```

```
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "None"
      }
    },
    "SECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Seconds"
      }
    },
    "MICROSECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Microseconds"
      }
    },
    "MILLISECONDS": {
      "target": "smithy.api#Unit",
      "traits": {

      "smithy.api#enumValue": "Milliseconds"
      }
    },
    "BYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
```

```
        "TERABYTES": {
            "target":
"smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Terabytes"
            }
        },
        "BITS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Bits"
            }
        },
        "KILOBITS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Kilobits"
            }
        },
        "MEGABITS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Megabits"
            }
        },
        "GIGABITS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Gigabits"
            }
        },

    "TERABITS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Terabits"
            }
        },
        "PERCENT": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Percent"
            }
        },
        "BYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Bytes/Second"
```

```
      }
    },
    "KILOBYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes/Second"
      }
    },
    "MEGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {

  "smithy.api#enumValue": "Megabytes/Second"
      }
    },
    "GIGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes/Second"
      }
    },
    "TERABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes/Second"
      }
    },
    "BITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits/Second"
      }
    },
    "KILOBITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobits/Second"
      }

},
    "MEGABITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits/Second"
      }
    },
    "GIGABITS_PER_SECOND": {
      "target": "smithy.api#Unit",
```

```
        "traits": {
          "smithy.api#enumValue": "Gigabits/Second"
        }
      },
      "TERABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits/Second"
        }
      },
      "COUNT_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Count/Second"
        }
      }
    }
  }
},
"com.amazonaws.licensemanager#EntitlementUsage": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement usage name.</p>",
          "smithy.api#required": {}
        }
      },
      "ConsumedValue": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Resource usage consumed.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxCount": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
        }
      },
      "Unit": {
        "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
         "traits": {
          "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
          "smithy.api#required": {}
        }
      }
```

```
      },
      "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementUsage"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
      },

    "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Extends the expiration date for license
consumption.</p>"
      }
    },
```

```
"com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
  "type": "structure",
  "members": {
    "LicenseConsumptionToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>License consumption token.</p>",
        "smithy.api#required": {}
      }
    },
    "DryRun": {
      "target": "com.amazonaws.licensemanager#Boolean",
      "traits": {
        "smithy.api#default": false,
        "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
  "type": "structure",
  "members": {
    "LicenseConsumptionToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>License consumption token.</p>"
      }
    },
    "Expiration": {
      "target": "com.amazonaws.licensemanager#ISO8601DateTime",
      "traits": {
        "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#FailedDependencyException": {
  "type": "structure",
  "members": {
    "Message": {
      "target": "com.amazonaws.licensemanager#Message"
```

```
      },
      "ErrorCode": {
        "target": "com.amazonaws.licensemanager#String"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "FailedDependency",
        "httpResponseCode": 424
      },
      "smithy.api#documentation": "<p>A dependency required to run the API is missing.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 424
    }
  },
  "com.amazonaws.licensemanager#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#FilterName",
        "traits": {
          "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanager#FilterValues",
        "traits": {
          "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a\n        describe operation. Filters can be used to match a set of resources by specific criteria,\n        such as
tags, attributes, or IDs.</p>"
    }
  },
  "com.amazonaws.licensemanager#FilterLimitExceededException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits":
{
      "aws.protocols#awsQueryError": {
```

"code": "FilterLimitExceeded",
            "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Filter"
      }
    },
    "com.amazonaws.licensemanager#FilterName": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#FilterValue": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#FilterValues": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#FilterValue",
        "traits": {

"smithy.api#xmlName": "item"
        }
      }
    },
    "com.amazonaws.licensemanager#Filters": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Filter",
        "traits": {
          "smithy.api#xmlName": "item"
        }
      }
    },
    "com.amazonaws.licensemanager#GetAccessToken": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
      },
      "errors": [

```
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target":
  "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets a temporary access token to use with
  AssumeRoleWithWebIdentity. Access tokens\n          are valid for one hour.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenRequest": {
          "type": "structure",
          "members": {
            "Token": {
              "target": "com.amazonaws.licensemanager#TokenString",
              "traits": {
                "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
                "smithy.api#required": {}
              }
            },
            "TokenProperties": {
              "target": "com.amazonaws.licensemanager#MaxSize3StringList",
              "traits": {
                "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
  token.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenResponse": {
          "type": "structure",
          "members": {
            "AccessToken": {
```

```
            "target": "com.amazonaws.licensemanager#TokenString",
            "traits": {
                "smithy.api#documentation": "<p>Temporary access token.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#GetGrant": {
    "type": "operation",

    "input": {
        "target": "com.amazonaws.licensemanager#GetGrantRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#GetGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },

        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
    }
},
"com.amazonaws.licensemanager#GetGrantRequest": {
```

```
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrantResponse": {
      "type": "structure",
      "members": {
        "Grant": {
          "target": "com.amazonaws.licensemanager#Grant",
          "traits": {
            "smithy.api#documentation": "<p>Grant details.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
```

```
{
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
    }
},
"com.amazonaws.licensemanager#GetLicenseConfiguration": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
    },

    "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license
```

```
configuration.</p>"
              }
          },
          "com.amazonaws.licensemanager#GetLicenseConfigurationRequest":
  {
              "type": "structure",
              "members": {
                  "LicenseConfigurationArn": {
                      "target": "com.amazonaws.licensemanager#String",
                      "traits": {
                          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
  configuration.</p>",
                          "smithy.api#required": {}
                      }
                  }
              },
              "traits": {
                  "smithy.api#input": {}
              }
          },
          "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
              "type": "structure",
              "members": {
                  "LicenseConfigurationId": {
                      "target": "com.amazonaws.licensemanager#String",
                      "traits": {
                          "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
                      }
                  },
                  "LicenseConfigurationArn":
  {
                      "target": "com.amazonaws.licensemanager#String",
                      "traits": {
                          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
                      }
                  },
                  "Name": {
                      "target": "com.amazonaws.licensemanager#String",
                      "traits": {
                          "smithy.api#documentation": "<p>Name of the license configuration.</p>"
                      }
                  },
                  "Description": {
                      "target": "com.amazonaws.licensemanager#String",
                      "traits": {
                          "smithy.api#documentation": "<p>Description of the license configuration.</p>"
                      }
                  },
```

"LicenseCountingType": {
    "target": "com.amazonaws.licensemanager#LicenseCountingType",
    "traits": {
        "smithy.api#documentation":
"<p>Dimension for which the licenses are counted.</p>"
    }
},
"LicenseRules": {
    "target": "com.amazonaws.licensemanager#StringList",
    "traits": {
        "smithy.api#documentation": "<p>License rules.</p>"
    }
},
"LicenseCount": {
    "target": "com.amazonaws.licensemanager#BoxLong",
    "traits": {
        "smithy.api#documentation": "<p>Number of available licenses.</p>"
    }
},
"LicenseCountHardLimit": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
    "traits": {
        "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
    }
},
"ConsumedLicenses": {
    "target":
"com.amazonaws.licensemanager#BoxLong",
    "traits": {
        "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
    }
},
"Status": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>License configuration status.</p>"
    }
},
"OwnerAccountId": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
    }
},
"ConsumedLicenseSummaryList": {
    "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
    "traits": {
        "smithy.api#documentation": "<p>Summaries

of the licenses consumed by resources.</p>"
                    }
                },
                "ManagedResourceSummaryList": {
                    "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
                    "traits": {
                        "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
                    }
                },
                "Tags": {
                    "target": "com.amazonaws.licensemanager#TagList",
                    "traits": {
                        "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
                    }
                },
                "ProductInformationList": {
                    "target": "com.amazonaws.licensemanager#ProductInformationList",
                    "traits": {
                        "smithy.api#documentation": "<p>Product information.</p>"
                    }
                },
                "AutomatedDiscoveryInformation":
{
                    "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
                    "traits": {
                        "smithy.api#documentation": "<p>Automated discovery information.</p>"
                    }
                },
                "DisassociateWhenNotFound": {
                    "target": "com.amazonaws.licensemanager#BoxBoolean",
                    "traits": {
                        "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                    }
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTask": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"

```
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
            "type": "structure",
            "members": {
                "LicenseConversionTaskId": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
            "type": "structure",
            "members": {
                "LicenseConversionTaskId": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
                    }
                },
                "ResourceArn": {
```

```
                "target": "com.amazonaws.licensemanager#String",
                 "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
                }
            },
            "SourceLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license type converted from.</p>"
                }
            },
            "DestinationLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanager#String",

       "traits": {
                    "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
                }
            },
            "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                }
            },
            "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Amount of time to complete the license type
conversion.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was
```

completed.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },

{

                "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
              {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              },
              {
                "target": "com.amazonaws.licensemanager#ValidationException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"

            }
          },

```
"com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
    "type": "structure",
    "members": {
        "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
    "type": "structure",
    "members": {
        "ReportGenerator": {
            "target": "com.amazonaws.licensemanager#ReportGenerator",
            "traits": {
                "smithy.api#documentation": "<p>A report generator that creates
periodic reports about your license configurations.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#GetLicenseRequest": {
    "type": "structure",
    "members": {
        "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                "smithy.api#required": {}
            }
        },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>License version.</p>"
            }
        }
    },
    "traits": {
```

```
            "smithy.api#input": { }
        }
    },

    "com.amazonaws.licensemanager#GetLicenseResponse": {
        "type": "structure",
        "members": {
            "License": {
                "target": "com.amazonaws.licensemanager#License",
                "traits": {
                    "smithy.api#documentation": "<p>License details.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": { }
        }
    },
    "com.amazonaws.licensemanager#GetLicenseUsage": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {

              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
        "type": "structure",
        "members": {
          "LicenseUsage": {
            "target": "com.amazonaws.licensemanager#LicenseUsage",
            "traits": {
              "smithy.api#documentation": "<p>License usage details.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
        },
         "output": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "S3BucketArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"
            }
          },
          "SnsTopicArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
            }
          },
          "OrganizationConfiguration": {
            "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
            "traits": {
              "smithy.api#documentation":
"<p>Indicates whether Organizations is integrated with License Manager for\n        cross-account discovery.</p>"
            }
          },
          "EnableCrossAccountsDiscovery": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
            }
          },
          "LicenseManagerResourceShareArn": {
```

```
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n        provides member accounts with access to this share.</p>"
                    }
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },

    "com.amazonaws.licensemanager#Grant": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant name.</p>",
                    "smithy.api#required": {}
                }
            },
            "ParentArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Parent ARN.</p>",
                    "smithy.api#required": {}
                }

            },
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>License ARN.</p>",
                    "smithy.api#required": {}
                }
            },
            "GranteePrincipalArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The grantee principal ARN.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantStatus": {
          "target":
"com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Grant status reason.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation":
"<p>Granted operations.</p>",
            "smithy.api#required": {}
          }
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
            "smithy.api#documentation": "<p>The options specified for the grant.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a grant.</p>"
```

```
          }
        },
      "com.amazonaws.licensemanager#GrantList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Grant"
        }
      },
      "com.amazonaws.licensemanager#GrantStatus": {
        "type": "enum",
        "members": {
          "PENDING_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue":
"PENDING_WORKFLOW"
            }
          },
          "PENDING_ACCEPT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_ACCEPT"
            }
          },
          "REJECTED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "REJECTED"
            }
          },
          "ACTIVE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ACTIVE"
            }
          },
          "FAILED_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED_WORKFLOW"
            }
          },
          "DELETED": {
            "target":
"smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DELETED"
            }
```

```
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
      "type": "structure",
      "members": {
        "LicenseArn": {

          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>"
          }
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
          }
```

                },
                "Issuer":
{
                    "target": "com.amazonaws.licensemanager#IssuerDetails",
                    "traits": {
                        "smithy.api#documentation": "<p>Granted license issuer.</p>"
                    }
                },
                "HomeRegion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Home Region of the granted license.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#LicenseStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Granted license status.</p>"
                    }
                },
                "Validity": {
                    "target": "com.amazonaws.licensemanager#DatetimeRange",
                    "traits": {
                        "smithy.api#documentation": "<p>Date and time range during which the
granted license is valid, in ISO8601-UTC format.</p>"
                    }
                },
                "Beneficiary": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
                    }
                },
                "Entitlements": {
                    "target": "com.amazonaws.licensemanager#EntitlementList",
                    "traits": {
                        "smithy.api#documentation": "<p>License entitlements.</p>"
                    }
                },
                "ConsumptionConfiguration": {
                    "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
                    }
                },
                "LicenseMetadata": {

```
      "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Granted license metadata.</p>"
                }
            },
            "CreateTime": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Creation time of the granted license.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Version of the granted license.</p>"
                }
            },
            "ReceivedMetadata": {
                "target": "com.amazonaws.licensemanager#ReceivedMetadata",
                "traits": {
                    "smithy.api#documentation": "<p>Granted license received metadata.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
        }
    },
    "com.amazonaws.licensemanager#GrantedLicenseList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#GrantedLicense"
        }
    },
    "com.amazonaws.licensemanager#ISO8601DateTime": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
        }
    },
    "com.amazonaws.licensemanager#Integer": {
        "type": "integer"
    },
```

```
            "com.amazonaws.licensemanager#InvalidParameterValueException":
    {
            "type": "structure",
            "members": {
              "Message": {
                "target": "com.amazonaws.licensemanager#Message"
              }
            },
            "traits": {
              "aws.protocols#awsQueryError": {
                "code": "InvalidParameterValueProvided",
                "httpResponseCode": 400
              },
              "smithy.api#documentation": "<p>One or more parameter values are not valid.</p>",
              "smithy.api#error": "client",
              "smithy.api#httpError": 400
            }
        },
        "com.amazonaws.licensemanager#InvalidResourceStateException": {
            "type": "structure",
            "members": {
              "Message": {
                "target": "com.amazonaws.licensemanager#Message"
              }
            },
            "traits": {
              "aws.protocols#awsQueryError": {

            "code": "InvalidResourceState",
                "httpResponseCode": 400
              },
              "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
              "smithy.api#error": "client",
              "smithy.api#httpError": 400
            }
        },
        "com.amazonaws.licensemanager#InventoryFilter": {
            "type": "structure",
            "members": {
              "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Name of the filter.</p>",
                  "smithy.api#required": {}
                }
              },
```

```json
        "Condition": {
          "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
          "traits": {
            "smithy.api#documentation": "<p>Condition of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An inventory filter.</p>"
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterCondition": {
      "type": "enum",
      "members": {
        "EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EQUALS"
          }
        },
        "NOT_EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "NOT_EQUALS"
          }
        },
        "BEGINS_WITH": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "BEGINS_WITH"
          }
        },
        "CONTAINS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CONTAINS"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterList": {
```

```
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#InventoryFilter"
          }
        },
        "com.amazonaws.licensemanager#Issuer": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Issuer name.</p>",
                "smithy.api#required": {}
              }
            },
            "SignKey": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
          }
        },
        "com.amazonaws.licensemanager#IssuerDetails": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits":
{
                "smithy.api#documentation": "<p>Issuer name.</p>"
              }
            },
            "SignKey": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
              }
            },
            "KeyFingerprint": {
              "target": "com.amazonaws.licensemanager#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#License": {

        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>"
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product name.</p>"
            }
          },
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation":
"<p>Product SKU.</p>"
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#IssuerDetails",
            "traits": {
              "smithy.api#documentation": "<p>License issuer.</p>"
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region of the license.</p>"
```

```
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {

      "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-UTC
format.</p>"
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>"
          }
        },
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
          }

    },
        "LicenseMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>License metadata.</p>"
          }
        },
        "CreateTime": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>License creation time.</p>"
          }
        },
        "Version": {
```

```
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License version.</p>"
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseConfiguration":
{
        "type": "structure",
        "members": {
            "LicenseConfigurationId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
                }
            },
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
                }
            },
            "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Name of the license configuration.</p>"
                }
            },
            "Description": {
                "target": "com.amazonaws.licensemanager#String",
                 "traits": {
                    "smithy.api#documentation": "<p>Description of the license configuration.</p>"
                }
            },
            "LicenseCountingType": {
                "target": "com.amazonaws.licensemanager#LicenseCountingType",
                "traits": {
                    "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
                }
            },
            "LicenseRules": {
                "target": "com.amazonaws.licensemanager#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>License rules.</p>"
```

```
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license
configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
          }
        },
        "Status": {

"target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the license configuration.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"
          }
        },
        "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
          }
        },
        "ManagedResourceSummaryList": {
```

```
        "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
        "traits":
{
            "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
        }
      },
      "ProductInformationList": {
        "target": "com.amazonaws.licensemanager#ProductInformationList",
        "traits": {
          "smithy.api#documentation": "<p>Product information.</p>"
        }
      },
      "AutomatedDiscoveryInformation": {
        "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
        "traits": {
          "smithy.api#documentation": "<p>Automated discovery information.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n       consumed and enforced by License Manager. Components include specifications for
the license\n
      type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n        Dedicated
Instance, Dedicated Host, or all of these), host affinity (how long a VM\n       must be associated with a host), and
the number of licenses purchased and used.</p>"
    }
  },
  "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Type of server resource.</p>"
        }
      },

      "ResourceOwnerId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
```

consuming licenses.</p>"
                }
              },
              "AssociationTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                  "smithy.api#documentation": "<p>Time when the license configuration was associated with the
resource.</p>"
                }
              },
              "AmiAssociationScope": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
                }
              }
            },
            "traits": {

  "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
            }
          },
          "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
            "type": "list",
            "member": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
            }
          },
          "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
            "type": "enum",
            "members": {
              "AVAILABLE": {
                "target": "smithy.api#Unit",
                "traits": {
                  "smithy.api#enumValue": "AVAILABLE"
                }
              },
              "DISABLED": {
                "target": "smithy.api#Unit",
                "traits": {
                  "smithy.api#enumValue": "DISABLED"
                }
              }
            }
          },
          "com.amazonaws.licensemanager#LicenseConfigurationUsage": {

```
"type": "structure",
"members": {
  "ResourceArn": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
      "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
    }
  },
  "ResourceType": {
    "target": "com.amazonaws.licensemanager#ResourceType",
    "traits": {
      "smithy.api#documentation": "<p>Type of resource.</p>"
    }
  },
  "ResourceStatus": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
      "smithy.api#documentation": "<p>Status of the resource.</p>"
    }
  },
  "ResourceOwnerId": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {

      "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
    }
  },
  "AssociationTime": {
    "target": "com.amazonaws.licensemanager#DateTime",
    "traits": {
      "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
    }
  },
  "ConsumedLicenses": {
    "target": "com.amazonaws.licensemanager#BoxLong",
    "traits": {
      "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
  }
},
"com.amazonaws.licensemanager#LicenseConfigurationUsageList":
```

```json
    {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
        }
    },
    "com.amazonaws.licensemanager#LicenseConfigurations": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#LicenseConfiguration"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionContext": {
        "type": "structure",
        "members": {
            "UsageOperation": {
                "target": "com.amazonaws.licensemanager#UsageOperation",
                "traits": {
                    "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample
 data: usage operation by platform\n      </a>\n        </p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
                }
            },
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n
    conversion task.</p>"
                }
            },
            "SourceLicenseContext": {
```

          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
          }
        },
        "DestinationLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the conversion task.</p>"

    }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
          }
        },
        "StartTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>The time the conversion task was started at.</p>"
          }
        },
        "LicenseConversionTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
          }
        },
        "EndTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
           "traits": {
            "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTaskId": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
        "type": "enum",
        "members": {
          "IN_PROGRESS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "IN_PROGRESS"
            }

          },
          "SUCCEEDED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "SUCCEEDED"
            }
          },
          "FAILED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTasks": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTask"
        }
      },
      "com.amazonaws.licensemanager#LicenseCountingType": {
        "type": "enum",
        "members": {
          "VCPU": {
            "target": "smithy.api#Unit",
            "traits": {
```

```
                    "smithy.api#enumValue": "vCPU"
                }
            },

        "INSTANCE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Instance"
                }
            },
            "CORE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Core"
                }
            },
            "SOCKET": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Socket"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#LicenseDeletionStatus": {
        "type": "enum",
        "members": {
            "PENDING_DELETE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "PENDING_DELETE"
                }
            },
            "DELETED":
{
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DELETED"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#LicenseList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#License"
        }
    },
```

```
"com.amazonaws.licensemanager#LicenseOperationFailure": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
        },
        "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
                "smithy.api#documentation":
"<p>Resource type.</p>"
            }
        },
        "ErrorMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Error message.</p>"
            }
        },
        "FailureTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
                "smithy.api#documentation": "<p>Failure time.</p>"
            }
        },
        "OperationName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Name of the operation.</p>"
            }
        },
        "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation":
"<p>ID of the Amazon Web Services account that owns the resource.</p>"
            }
        },
        "OperationRequestedBy": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
            }
        },
        "MetadataList": {
```

```
                    "target": "com.amazonaws.licensemanager#MetadataList",
                    "traits": {
                        "smithy.api#documentation": "<p>Reserved.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
            }
        },
        "com.amazonaws.licensemanager#LicenseOperationFailureList": {
            "type": "list",
            "member": {
                "target":
 "com.amazonaws.licensemanager#LicenseOperationFailure"
            }
        },
        "com.amazonaws.licensemanager#LicenseSpecification": {
            "type": "structure",
            "members": {
                "LicenseConfigurationArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                        "smithy.api#required": {}
                    }
                },
                "AmiAssociationScope": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Details for associating a license configuration
 with a resource.</p>"
            }
        },
        "com.amazonaws.licensemanager#LicenseSpecifications": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#LicenseSpecification"
            }
        },
        "com.amazonaws.licensemanager#LicenseStatus": {
```

```
      "type": "enum",
      "members": {
        "AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "PENDING_AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_AVAILABLE"
          }
        },
        "DEACTIVATED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DEACTIVATED"
          }
        },
        "SUSPENDED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUSPENDED"
          }
        },
        "EXPIRED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EXPIRED"
          }
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseUsage": {
      "type":
```

```
          "structure",
              "members": {
                "EntitlementUsages": {
                   "target": "com.amazonaws.licensemanager#EntitlementUsageList",
                   "traits": {
                      "smithy.api#documentation": "<p>License entitlement usages.</p>"
                   }
                }
              },
              "traits": {
                "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"
              }
          },
          "com.amazonaws.licensemanager#LicenseUsageException": {
              "type": "structure",
              "members": {
                "Message": {
                   "target": "com.amazonaws.licensemanager#Message"
                }
              },
              "traits": {
                "aws.protocols#awsQueryError": {
                   "code": "LicenseUsageFailure",
                   "httpResponseCode": 412
                },
                "smithy.api#documentation": "<p>You do not have
enough licenses available to support a new resource launch.</p>",
                "smithy.api#error": "client",
                "smithy.api#httpError": 412
              }
          },
          "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
              "type": "operation",
              "input": {
                "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
              },
              "output": {
                "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
              },
              "errors": [
                {
                   "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                   "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                   "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
```

          },
          {

        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the resource associations for the specified license
configuration.</p>\n   <p>Resource associations need not consume licenses from a license configuration. \n
For example, an AMI or a stopped instance might not consume a license (depending on \n   the license
rules).</p>"
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse":

```
    {
        "type": "structure",
        "members": {
            "LicenseConfigurationAssociations": {
                "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListDistributedGrants": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
```

```
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
        "type": "structure",
        "members": {
          "GrantArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n                <code>LicenseArn</code>\n            </p>\n

 </li>\n          <li>\n            <p>\n                <code>GrantStatus</code>\n            </p>\n          </li>\n
<li>\n            <p>\n                <code>GranteePrincipalARN</code>\n            </p>\n          </li>\n          <li>\n
     <p>\n                <code>ProductSKU</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
     <code>LicenseIssuerName</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
```

```
        "type": "structure",
        "members": {
          "Grants": {
            "target": "com.amazonaws.licensemanager#GrantList",
            "traits": {
              "smithy.api#documentation": "<p>Distributed grant details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": { }
        }
    },

    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
        }
```

```
        },
        "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse": {
          "type": "structure",
          "members": {
            "LicenseOperationFailureList": {
              "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
              "traits": {
                "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {

                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },

    {

            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configurations for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
            }
          },
          "MaxResults":

 {

            "target": "com.amazonaws.licensemanager#BoxInteger",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n    are supported:</p>\n    <ul>\n      <li>\n        <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n          Possible
 values are <code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n
Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n
      <p>\n          <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n          Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n        <p>\n
<code>usagelimitExceeded</code> - A Boolean value that indicates whether the available licenses have been
exceeded. \n          Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
</li>\n      </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
        "type": "structure",
        "members": {
          "LicenseConfigurations": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurations",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
```

```
        "smithy.api#output": { }
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
      },
      "output": {

      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {

"type": "structure",
      "members": {
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
```

```
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>\n        Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n     </p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTasks": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
         },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n        <ul>\n          <li>\n            <p>\n              <code>LicenseConfigurationArn</code>\n
</p>\n
        </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
```

```
        },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
      "type": "structure",
      "members": {
        "ReportGenerators": {
          "target": "com.amazonaws.licensemanager#ReportGeneratorList",
          "traits": {
            "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
      },
      "errors":
[
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
```

```
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {

 "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse":
{
          "type": "structure",
          "members": {
            "LicenseSpecifications": {
              "target": "com.amazonaws.licensemanager#LicenseSpecifications",
              "traits": {
                "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
              }
            },
            "NextToken": {
```

```
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsRequest":
    {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
```

```
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits":
{
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
        "type": "structure",
        "members": {
            "Licenses": {
                "target": "com.amazonaws.licensemanager#LicenseList",
                "traits": {
                    "smithy.api#documentation": "<p>License details.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenses": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicensesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListLicensesResponse"
        },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists
the licenses for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n                <code>Beneficiary</code>\n            </p>\n
  </li>\n          <li>\n            <p>\n                <code>ProductSKU</code>\n            </p>\n          </li>\n
<li>\n
            <p>\n                <code>Fingerprint</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
          <code>Status</code>\n            </p>\n          </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```json
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicensesResponse":
{
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#LicenseList",
            "traits": {
              "smithy.api#documentation": "<p>License details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedGrants": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
```

```
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>Lists grants that are received. Received grants are grants created while specifying the\n       recipient as this
Amazon Web Services account, your organization, or an organizational unit\n       (OU) to which this member
account belongs.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received
license.</p>",
                "smithy.api#required": {}
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n      <ul>\n         <li>\n            <p>\n               <code>ParentArn</code>\n            </p>\n
      </li>\n         <li>\n            <p>\n               <code>GranteePrincipalArn</code>\n            </p>\n
</li>\n      </ul>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
```

```
                }
            },
            "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse": {
                "type": "structure",
                "members": {
                    "Grants": {
                        "target": "com.amazonaws.licensemanager#GrantList",
                        "traits": {
                            "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
                        }
                    },
                    "NextToken": {
                        "target": "com.amazonaws.licensemanager#String",

    "traits": {
                            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                        }
                    }
                },
                "traits": {
                    "smithy.api#output": {}
                }
            },
            "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
                "type": "structure",
                "members": {
                    "GrantArns": {
                        "target": "com.amazonaws.licensemanager#ArnList",
                        "traits": {
                            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
                        }
                    },
                    "Filters": {
                        "target": "com.amazonaws.licensemanager#FilterList",
                        "traits": {
                            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>ProductSKU</code>\n
       </p>\n          </li>\n          <li>\n            <p>\n              <code>LicenseIssuerName</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>LicenseArn</code>\n            </p>\n
</li>\n          <li>\n            <p>\n              <code>GrantStatus</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>GranterAccountId</code>\n            </p>\n          </li>\n        </ul>"
                        }
                    },
                    "NextToken": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                        }
```

```
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",

    "traits": {
                  "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
          "type": "structure",
          "members": {
            "Grants": {
              "target": "com.amazonaws.licensemanager#GrantList",
              "traits": {
                "smithy.api#documentation": "<p>Received grant details.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output":
{}
          }
        },
        "com.amazonaws.licensemanager#ListReceivedLicenses": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```json
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
         },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists received licenses.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
         },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
```

```
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
          "type": "structure",
          "members":
    {
            "Filters": {
              "target": "com.amazonaws.licensemanager#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
    supported:</p>\n      <ul>\n         <li>\n            <p>\n               <code>Beneficiary</code>\n            </p>\n
         </li>\n         <li>\n            <p>\n               <code>ProductSKU</code>\n            </p>\n         </li>\n
    </ul>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",
              "traits": {

                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
          "type": "structure",
          "members": {
            "Licenses": {
              "target": "com.amazonaws.licensemanager#GrantedLicenseList",
              "traits": {
                "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
              }
            },
```

```
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {

    "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
    "type": "structure",
    "members": {
      "LicenseArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n     <ul>\n        <li>\n          <p>\n            <code>ProductSKU</code>\n          </p>\n
        </li>\n        <li>\n          <p>\n            <code>Status</code>\n          </p>\n        </li>\n
<li>\n
          <p>\n            <code>Fingerprint</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
        <code>IssuerName</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>Beneficiary</code>\n          </p>\n        </li>\n     </ul>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#MaxSize100",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      }
    },
    "traits":
```

```json
{
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
        "type": "structure",
        "members": {
            "Licenses": {
                "target": "com.amazonaws.licensemanager#GrantedLicenseList",
                "traits": {
                    "smithy.api#documentation": "<p>Received license details.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListResourceInventory": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#FailedDependencyException"
            },
            {
                "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
```

```
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target":
"com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target":
"com.amazonaws.licensemanager#InventoryFilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n      are supported:</p>\n      <ul>\n        <li>\n          <p>\n            <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n          <p>\n
   <code>application_name</code> - The name of the application. \n            Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n        </li>\n        <li>\n          <p>\n
   <code>license_included</code> - The type of license included. \n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.\n            Possible values are <code>sql-server-
enterprise</code>
| \n            <code>sql-server-standard</code> | \n            <code>sql-server-web</code> | \n
<code>windows-server-datacenter</code>.</p>\n        </li>\n        <li>\n          <p>\n
<code>platform</code> - The platform of the resource.\n            Logical operators are <code>EQUALS</code> |
<code>BEGINS_WITH</code>.</p>\n        </li>\n        <li>\n          <p>\n
<code>resource_id</code> - The ID of the resource. \n            Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n          <p>\n
<code>tag:<key></code> - The key/value combination of a tag assigned \n            to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n            <code>EQUALS</code> |
```

```
<code>NOT_EQUALS</code> (cross account).</p>\n          </li>\n      </ul>"

  }
            }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
    "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
        "type": "structure",
        "members": {
          "ResourceInventoryList": {
            "target": "com.amazonaws.licensemanager#ResourceInventoryList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the resources.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
    "com.amazonaws.licensemanager#ListTagsForResource": {
        "type":
"operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
```

```
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the
tags for the specified license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the tags.</p>"

          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListTokens": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListTokensRequest"
      },
```

```
      "output": {
        "target": "com.amazonaws.licensemanager#ListTokensResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {

  "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists your tokens.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTokensRequest": {
      "type": "structure",
      "members": {
        "TokenIds": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>Token IDs.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n          <li>\n            <p>\n                <code>LicenseArns</code>\n             </p>\n          </li>\n
</ul>"

        }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
```

```
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListTokensResponse": {
      "type": "structure",
      "members": {
        "Tokens": {
          "target": "com.amazonaws.licensemanager#TokenList",
          "traits": {
            "smithy.api#documentation":
"<p>Received token details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
```

```
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n      consumption details by resource at a selected point in time. Use this action to audit the\n      current
license consumption for any license inventory and configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest":
{
    "type": "structure",
    "members": {
      "LicenseConfigurationArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"

        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#Filters",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n          are supported:</p>\n          <ul>\n          <li>\n              <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n          <li>\n              <p>\n
    <code>resourceType</code> - The resource type (<code>EC2_INSTANCE</code> |
<code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n          <li>\n              <p>\n
    <code>resourceAccount</code>
 - The ID of the account that owns the resource. \n          Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n          </li>\n          </ul>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurationUsageList": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Location": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Long": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#ManagedResourceSummary": {
      "type": "structure",
      "members": {
```

```json
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
        }
      },
      "AssociationCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"

}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
    }
  },
  "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
    }
  },
  "com.amazonaws.licensemanager#MaxSize100": {
    "type": "integer",
    "traits": {
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  },
  "com.amazonaws.licensemanager#MaxSize3StringList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 3
      }

  }
  },
  "com.amazonaws.licensemanager#Message": {
    "type": "string"
```

```
        },
        "com.amazonaws.licensemanager#Metadata": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The key name.</p>"
              }
            },
            "Value": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
          }
        },
        "com.amazonaws.licensemanager#MetadataList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Metadata"
          }
        },
        "com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#Options": {
          "type": "structure",
          "members": {
            "ActivationOverrideBehavior": {
              "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
              "traits": {
                "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
```

activating

 a grant.\n   Activation options can only be used with granted licenses sourced from the Amazon Web Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n <code>Status</code> parameter.</p>\n   \n   \n   <p>As a license administrator, you can optionally specify an\n   <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n \n   \n   <p>As a grantor, you can optionally specify an\n <code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n   account in your organization.</p>\n   \n   \n   <p>As a grantee, if the grantor creating the distributed grant doesnt specify an\n   <code>ActivationOverrideBehavior</code>, you can optionally specify one when you\n   are activating the grant.</p>\n

    \n   \n   <dl>\n   <dt>DISTRIBUTED_GRANTS_ONLY</dt>\n   <dd>\n <p>Use this value to activate a grant without replacing any member accounts\n   active grants for the same product.</p>\n   </dd>\n   <dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n   <dd>\n   <p>Use this value to activate a grant and disable other active grants in any\n   member accounts for the same product. This action will also replace their\n   previously activated grants with this activated grant.</p>\n   </dd>\n   </dl>"
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant, such as activation\n   override behavior. For more information, see Granted licenses in License Manager in the <i>License Manager User Guide</i>.</p>"
            }
          },
          "com.amazonaws.licensemanager#OrganizationConfiguration": {
            "type": "structure",
            "members": {
              "EnableIntegration": {
                "target": "com.amazonaws.licensemanager#Boolean",
                "traits": {
                  "smithy.api#default": false,
                  "smithy.api#documentation": "<p>Enables Organizations integration.</p>",
                  "smithy.api#required": {}
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
            }
          },
          "com.amazonaws.licensemanager#PrincipalArnList": {
            "type": "list",
            "member": {
              "target": "com.amazonaws.licensemanager#Arn"
            },
            "traits":

```
{
        "smithy.api#length": {
            "min": 1,
            "max": 1
        }
    }
},
"com.amazonaws.licensemanager#ProductInformation": {
    "type": "structure",
    "members": {
        "ResourceType": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Resource type. The possible values are
<code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                "smithy.api#required": {}
            }
        },
        "ProductInformationFilterList": {
            "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
            "traits": {
                "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code>
 which specifies the type of filter being declared, and a <code>ProductInformationFilterValue</code> that specifies
the value to filter on. </p>\n        <p>Accepted values for <code>ProductInformationFilterName</code> are listed
here along with descriptions and valid options for <code>ProductInformationFilterComparator</code>. </p>\n
<p>The following filters and are supported when the resource type \n        is
<code>SSM_MANAGED</code>:</p>\n        <ul>\n        <li>\n            <p>\n            <code>Application
Name</code> - The name of the application.\n        Logical operator is <code>EQUALS</code>.</p>\n
</li>\n        <li>\n            <p>\n            <code>Application Publisher</code> - The publisher of the
application.\n        Logical operator is <code>EQUALS</code>.</p>\n        </li>\n        <li>\n
<p>\n            <code>Application Version</code> - The
 version of the application.\n        Logical operator is <code>EQUALS</code>.</p>\n        </li>\n
<li>\n            <p>\n            <code>Platform Name</code> - The name of the platform.\n        Logical
operator is <code>EQUALS</code>.</p>\n        </li>\n        <li>\n            <p>\n            <code>Platform
Type</code> - The platform type.\n        Logical operator is <code>EQUALS</code>.</p>\n        </li>\n
 <li>\n            <p>\n            <code>Tag:key</code> - The key of a tag attached to an Amazon Web Services
resource you wish to exclude from automated discovery. Logical operator is <code>NOT_EQUALS</code>.  The
key for your tag must be appended to <code>Tag:</code> following the example: <code>Tag:name-of-your-
key</code>. <code>ProductInformationFilterValue</code> is optional if you are not using values for the key.\n
 </p>\n        </li>\n        <li>\n
 <p>\n            <code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to
exclude from automated discovery.\n        Logical operator is <code>NOT_EQUALS</code>.</p>\n
</li>\n        <li>\n            <p>\n            <code>License Included</code> - The type of license included.\n
 Logical operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n        Possible values
are: <code>sql-server-enterprise</code> | \n            <code>sql-server-standard</code> | \n            <code>sql-
server-web</code> | \n            <code>windows-server-datacenter</code>.</p>\n        </li>\n        </ul>\n
```

<p>The following filters and logical operators are supported when the resource type\n   is
<code>RDS</code>:</p>\n   \n   \n   <p>\n   <code>Engine Edition</code> -
The edition of the database engine.\n   Logical
 operator is <code>EQUALS</code>.\n   Possible values are: <code>oracle-ee</code> | <code>oracle-
se</code> | <code>oracle-se1</code> | <code>oracle-se2</code>.</p>\n   \n   \n   <p>\n
          <code>License Pack</code> - The license pack.\n   Logical operator is
<code>EQUALS</code>.\n   Possible values are: <code>data guard</code> | \n   <code>diagnostic
pack sqlt</code> | \n   <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n
<code>olap</code>.</p>\n   \n   ",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#ProductInformationFilter": {
          "type": "structure",
          "members": {
            "ProductInformationFilterName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Filter name.</p>",
                "smithy.api#required": {}
              }
            },
            "ProductInformationFilterValue": {
              "target": "com.amazonaws.licensemanager#StringList",
              "traits": {
                "smithy.api#documentation": "<p>Filter value.</p>"
              }
            },
            "ProductInformationFilterComparator": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Logical operator.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation":
"<p>Describes product information filters.</p>"
          }
        },
        "com.amazonaws.licensemanager#ProductInformationFilterList": {
          "type": "list",

```
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformationFilter"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformation"
      }
    },
    "com.amazonaws.licensemanager#ProvisionalConfiguration": {
      "type": "structure",
      "members": {
        "MaxTimeToLiveInMinutes": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
            "smithy.api#required": {}

        }
          }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#RateLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "RateLimitExceeded",
          "httpResponseCode": 429
        },
        "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 429
      }
    },
    "com.amazonaws.licensemanager#ReceivedMetadata": {
      "type": "structure",
      "members": {
```

```json
        "ReceivedStatus": {
            "target": "com.amazonaws.licensemanager#ReceivedStatus",
            "traits": {
                "smithy.api#documentation": "<p>Received status.</p>"
            }
        },
        "ReceivedStatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
                "smithy.api#documentation": "<p>Received status reason.</p>"
            }
        },
        "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
                "smithy.api#documentation": "<p>Allowed operations.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
    }
},
"com.amazonaws.licensemanager#ReceivedStatus":
{
    "type": "enum",
    "members": {
        "PENDING_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "PENDING_WORKFLOW"
            }
        },
        "PENDING_ACCEPT": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "PENDING_ACCEPT"
            }
        },
        "REJECTED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "REJECTED"
            }
        },
        "ACTIVE": {
            "target": "smithy.api#Unit",
            "traits": {
```

```
              "smithy.api#enumValue": "ACTIVE"
            }
          },
          "FAILED_WORKFLOW": {

            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED_WORKFLOW"
            }
          },
          "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DELETED"
            }
          },
          "DISABLED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DISABLED"
            }
          },
          "WORKFLOW_COMPLETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "WORKFLOW_COMPLETED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#RedirectException": {
        "type": "structure",
        "members": {
          "Location":
{
            "target": "com.amazonaws.licensemanager#Location",
            "traits": {
              "smithy.api#httpHeader": "Location"
            }
          },
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 308
```

```
        }
      },
      "com.amazonaws.licensemanager#RejectGrant": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#RejectGrantRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#RejectGrantResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#RejectGrantRequest":
{
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#RejectGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {

  "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#RenewType": {
        "type": "enum",
        "members": {
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "WEEKLY": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Weekly"
            }
          },
          "MONTHLY": {
```

```json
        "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Monthly"
          }
        }
      }
    }
  },
  "com.amazonaws.licensemanager#ReportContext": {
    "type": "structure",
    "members": {
      "licenseConfigurationArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that
this generator reports on.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
    }
  },
  "com.amazonaws.licensemanager#ReportFrequency": {
    "type": "structure",
    "members":
 {
      "value": {
        "target": "com.amazonaws.licensemanager#Integer",
        "traits": {
          "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
        }
      },
      "period": {
        "target": "com.amazonaws.licensemanager#ReportFrequencyType",
        "traits": {
          "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
    }
  },
  "com.amazonaws.licensemanager#ReportFrequencyType": {
```

```
            "type": "enum",
            "members": {
                "DAY": {

                 "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "DAY"
                    }
                },
                "WEEK": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "WEEK"
                    }
                },
                "MONTH": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "MONTH"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#ReportGenerator": {
            "type": "structure",
            "members": {
                "ReportGeneratorName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Name of the report generator.</p>"
                    }
                },

        "ReportType": {
                    "target": "com.amazonaws.licensemanager#ReportTypeList",
                    "traits": {
                        "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
                    }
                },
                "ReportContext": {
                    "target": "com.amazonaws.licensemanager#ReportContext",
                    "traits": {
                        "smithy.api#documentation": "<p>License configuration type for this generator.</p>"
                    }
                },
                "ReportFrequency": {
                    "target": "com.amazonaws.licensemanager#ReportFrequency",
                    "traits": {
```

```
              "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
            }
          },
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",

    "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
            }
          },
          "LastRunStatus": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
            }
          },
          "LastRunFailureReason": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
            }
          },
          "LastReportGenerationTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
            }
          },
          "ReportCreatorAccount": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
          },
          "S3Location": {
            "target": "com.amazonaws.licensemanager#S3Location",
            "traits": {
              "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
            }
          },
```

```
        "CreateTime": {
          "target":
"com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the report was created.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportGenerator"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorName": {
      "type": "string",
      "traits": {
        "smithy.api#length":
{
          "min": 1,
          "max": 100
        }
      }
    },
    "com.amazonaws.licensemanager#ReportType": {
      "type": "enum",
      "members": {
        "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
          }
        },
        "LICENSE_CONFIGURATION_USAGE_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationUsageReport"
          }
```

```
                }
            }
        },
        "com.amazonaws.licensemanager#ReportTypeList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#ReportType"
            }
        },
        "com.amazonaws.licensemanager#ResourceInventory":
{
            "type": "structure",
            "members": {
                "ResourceId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the resource.</p>"
                    }
                },
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#ResourceType",
                    "traits": {
                        "smithy.api#documentation": "<p>Type of resource.</p>"
                    }
                },
                "ResourceArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
                    }
                },
                "Platform": {
                    "target": "com.amazonaws.licensemanager#String",

    "traits": {
                        "smithy.api#documentation": "<p>Platform of the resource.</p>"
                    }
                },
                "PlatformVersion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
                    }
                },
                "ResourceOwningAccountId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about a resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#ResourceInventoryList": {
        "type": "list",
        "member": {
          "target":
"com.amazonaws.licensemanager#ResourceInventory"
        }
      },
      "com.amazonaws.licensemanager#ResourceLimitExceededException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "ResourceLimitExceeded",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidResource.NotFound",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>The resource cannot be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
```

```
"com.amazonaws.licensemanager#ResourceType": {
    "type": "enum",
    "members": {
        "EC2_INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "EC2_INSTANCE"
            }
        },
        "EC2_HOST": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "EC2_HOST"
            }
        },
        "EC2_AMI": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "EC2_AMI"
            }
        },
        "RDS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "RDS"
            }
        },
        "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
            }
        }
    }
},
"com.amazonaws.licensemanager#S3Location": {
    "type": "structure",
    "members": {
        "bucket": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
            }

        },
        "keyPrefix": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
              "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
        }
      },
      "com.amazonaws.licensemanager#ServerInternalException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InternalError",
            "httpResponseCode": 500
          },
          "smithy.api#documentation": "<p>The server experienced an internal
error. Try again.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanager#SignedToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 4096
          }
        }
      },
      "com.amazonaws.licensemanager#StatusReasonMessage": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 400
          },
          "smithy.api#pattern": "^[\\s\\S]+$"
        }
      },
      "com.amazonaws.licensemanager#String": {
        "type": "string"
      },
```

```
        "com.amazonaws.licensemanager#StringList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#String"
            }
        },
        "com.amazonaws.licensemanager#Tag":
{
            "type": "structure",
            "members": {
                "Key": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Tag key.</p>"
                    }
                },
                "Value": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Tag value.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#TagKeyList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#String"
            }
        },
        "com.amazonaws.licensemanager#TagList": {
            "type": "list",
            "member":
{
                "target": "com.amazonaws.licensemanager#Tag"
            }
        },
        "com.amazonaws.licensemanager#TagResource": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#TagResourceRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#TagResourceResponse"
            },
```

```
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>One or more tags.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
```

```
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#TokenData": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>"
            }
          },
          "TokenType": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
            }
          },
          "LicenseArn": {

            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "ExpirationTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC format.</p>"
            }
          },
          "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
            "traits": {
              "smithy.api#documentation": "<p>Data specified by the caller.</p>"
            }
          },
          "RoleArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation":
"<p>Amazon Resource Names (ARN) of the roles included in the token.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes a token.</p>"
        }
    },
    "com.amazonaws.licensemanager#TokenList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#TokenData"
        }
    },
    "com.amazonaws.licensemanager#TokenString": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 4096
            },

    "smithy.api#pattern": "^\\S+$"
        }
    },
    "com.amazonaws.licensemanager#TokenType": {
        "type": "enum",
        "members": {
            "REFRESH_TOKEN": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "REFRESH_TOKEN"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
            "smithy.api#error": "client",
```

```
              "smithy.api#httpError": 400
            }
        },
        "com.amazonaws.licensemanager#UntagResource":
{
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#UntagResourceRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#UntagResourceResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
              {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Removes
the specified tags from the specified license configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#UntagResourceRequest": {
            "type": "structure",
            "members": {
              "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                  "smithy.api#required": {}
                }
              },
              "TagKeys": {
                "target": "com.amazonaws.licensemanager#TagKeyList",
                "traits": {
```

```
              "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#UntagResourceResponse":
{
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }

},
          "LicenseConfigurationStatus": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
            "traits": {
              "smithy.api#documentation": "<p>New status of the license configuration.</p>"
            }
          },
          "LicenseRules": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a
license\n        configuration is licenseAffinityToHost.</p>"
            }
          },
          "LicenseCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
            }
          },
          "LicenseCountHardLimit":
{
              "target": "com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
              }
          },
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New name of the license configuration.</p>"
            }
          },
          "Description": {
```

```
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
             "smithy.api#documentation": "<p>New description of the license configuration.</p>"
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {

             "smithy.api#documentation": "<p>New product information.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
             "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportGeneratorName": {
            "target": "com.amazonaws.licensemanager#ReportGeneratorName",
            "traits": {
              "smithy.api#documentation": "<p>Name of the report generator.</p>",
              "smithy.api#required": {}
            }
          },
          "Type": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
              "smithy.api#documentation": "<p>Type of reports to generate. The following report types are
supported:</p>\n        <ul>\n          <li>\n              <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n          </li>\n          <li>\n              <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n          </li>\n
</ul>",
```

```
            "smithy.api#required": {}
          }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>The report context.</p>",
            "smithy.api#required": {}
          }
        },
        "ReportFrequency": {
          "target": "com.amazonaws.licensemanager#ReportFrequency",
          "traits": {
            "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
            "smithy.api#required": {}
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientRequestToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the report generator.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
```

        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
          },
          {
            "target": "com.amazonaws.licensemanager#LicenseUsageException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits":
{
        "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n   <p>You can update the license specifications of AMIs, instances, and
hosts.\n   You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
              "smithy.api#required": {}
            }
          },
          "AddLicenseSpecifications":

```
{
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
               "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
            }
         },
         "RemoveLicenseSpecifications": {
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
               "smithy.api#documentation": "<p>ARNs of the license configurations to remove.</p>"
            }
         }
      },
      "traits": {
         "smithy.api#input": {}
      }
   },
   "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
         "smithy.api#output": {}
      }
   },
   "com.amazonaws.licensemanager#UpdateServiceSettings": {

"type": "operation",
      "input": {
         "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
      },
      "output": {
         "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
      },
      "errors": [
         {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
         },
         {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
         },
         {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
         {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
         },
         {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Updates
 License Manager settings for the current Region.</p>"
            }
        },
        "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
            "type": "structure",
            "members": {
                "S3BucketArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License Manager information is stored.</p>"
                    }
                },
                "SnsTopicArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
                    }
                },
                "OrganizationConfiguration": {
                    "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
                    }
                },
                "EnableCrossAccountsDiscovery": {
                    "target": "com.amazonaws.licensemanager#BoxBoolean",
                    "traits": {
                        "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
```

```
            },
      "com.amazonaws.licensemanager#UsageOperation": {
          "type": "string",
          "traits": {
            "smithy.api#length": {
               "min":
 0,

               "max": 50
            }
          }
        },
      "com.amazonaws.licensemanager#ValidationException": {
          "type": "structure",
          "members": {
            "Message": {
               "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        }
      }
}
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
```

```go
	if err != nil {
		return nil, err
	}

	out := result.(*CheckoutBorrowLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type CheckoutBorrowLicenseInput struct {

	// Unique, case-sensitive identifier that you
	provide to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// Digital signature method. The possible value is JSON Web Signature (JWS)
	// algorithm PS384. For more information, see RFC 7518 Digital Signature with
	// RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
	//
	// This member is required.
	DigitalSignatureMethod types.DigitalSignatureMethod

	// License entitlements. Partial checkouts are not supported.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Amazon Resource Name (ARN) of the license. The license must use the borrow
	// consumption configuration.
	//
	// This member is required.
	LicenseArn *string

	// Information about constraints.
	CheckoutMetadata []types.Metadata

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

	// Information about constraints.
```

CheckoutMetadata []types.Metadata

	// Allowed license entitlements.
	EntitlementsAllowed
	[]types.EntitlementData

	// Date and time at which the license checkout expires.
	Expiration *string

	// Date and time at which the license checkout is issued.
	IssuedAt *string

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License consumption token.
	LicenseConsumptionToken *string

	// Node ID.
	NodeId *string

	// Signed token.
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {

```go
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = addHTTPSignerV4Middleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err
	= stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
	middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
```

```go
  }
  return nil
 }

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  SigningName:   "license-manager",
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
```

```
// This member is required.
Beneficiary *string

// Unique, case-sensitive identifier that you provide to ensure
the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
```

```
	//
	// This member is required.
	Validity *types.DatetimeRange

	//
	Information about the license.
	LicenseMetadata []types.Metadata

	noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack);
	err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = addHTTPSignerV4Middleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
}
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return
err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
 return nil
}
```

```go
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		SigningName:   "license-manager",
		OperationName: "CreateLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
	if params == nil {
		params = &ListReceivedLicensesInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*ListReceivedLicensesOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type ListReceivedLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//
	- ProductSKU
	//   - Status
	//   - Fingerprint
	//   - IssuerName
```

```go
	//   - Beneficiary
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

	// Received license details.
	Licenses []types.GrantedLicense

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	if
	err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
```

```
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
    return err
    }
    if err = addHTTPSignerV4Middleware(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
    }
    if err = addClientUserAgent(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return
    err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
    middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    return nil
}
```

```go
func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  SigningName:  "license-manager",
  OperationName: "ListReceivedLicenses",
 }
}
```

1.26 aws-aws-sdk-go-v2-aws-protocol-eventstream 1.5.3

1.26.1 Available under license :

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil,
 err
 }
```

```go
	out := result.(*AssociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type AssociateLicenseInput struct {

	// The type of license to associate with the workspace.
	//
	// This member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to associate the license with.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

	// A structure containing data about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
	if err !=
nil {
		return err
	}
	err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
```

```
	}

		if err = addlegacyEndpointContextSetter(stack, options); err != nil {
	return err
		}
		if err = addSetLoggerMiddleware(stack, options); err != nil {
	return err
		}
		if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
	return err
		}
		if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
		}
		if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
		}
		if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
		}
		if err = addRetryMiddlewares(stack, options); err != nil {
	return err
		}
		if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
		}
		if err = awsmiddleware.AddRecordResponseTiming(stack);
	err != nil {
	return err
		}
		if err = addClientUserAgent(stack, options); err != nil {
	return err
		}
		if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
	return err
		}
		if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
		}
		if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
	return err
		}
		if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
	return err
		}
		if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
	return err
```

```go
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack,
options); err != nil {
return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:       region,
ServiceID:     ServiceID,
OperationName: "AssociateLicense",
}
}
```

AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.

```yaml
name: License Scan

on: [pull_request]

jobs:
 licensescan:
  name: License Scan
  runs-on: ubuntu-latest
  strategy:
   matrix:
    python-version: [3.9]

  steps:
   - name: Checkout target
    uses: actions/checkout@v2
    with:
     path: sdkbase
```

```
      ref: ${{ github.base_ref }}
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
       path: new-ref
       fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
       repository: nexB/scancode-toolkit
       path: scancode-toolkit
       fetch-depth: 1
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
       python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager


import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```go
// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary
 *string

 noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
```

```go
if err != nil {
	return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
	return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
	return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
	return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
err != nil {
	return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
	return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
	return err
}
if err = addClientUserAgent(stack, options); err != nil {
	return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
	return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
	return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
```

```go
			return
err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckInLicense",
 }
}
```

```json
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>The type of name to filter by.</p>"
                  }
              },
              "Values": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                  "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
                }
              },
              "Operator": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
                "traits": {
                  "smithy.api#documentation": "<p>An operator for filtering results.</p>"
                }
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
          }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
          "type": "list",
          "member": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
          }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {

            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
```

```
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n      Organizations integration has been enabled, and
 which Regions data will be aggregated from.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
          "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
          }
        },
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The Region in which License Manager displays
the aggregated data for Linux\n     subscriptions.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
          "type": "structure",
          "members": {
            "AmiId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
              }
            },
            "InstanceID": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
              }
            },
            "InstanceType": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
              }

            },
            "AccountID": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
              }
            },
            "Region": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
              }
            },
            "UsageOperation": {
              "target": "smithy.api#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
  more information, see\n     <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n     operation values</a> in the <i>License Manager User
Guide</i>.</p>"
          }
        },
        "ProductCode": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
          "traits": {
            "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n     operation values</a> in the <i>License Manager User Guide</i> .</p>"
          }
        },
        "LastUpdatedTime": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
          }
        },
        "SubscriptionName": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
          }
        },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
      ],

      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager Linux Subscriptions",
          "arnNamespace": "license-manager-linux-subscriptions",
          "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",

"x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
```

```
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The
 AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
              "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
```

```
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {

          "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
              "type": "error"
            },
            {
              "conditions": [],
              "type": "tree",
              "rules": [
                {

              "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                  "type": "error"
                },
                {
```

```
                                 "conditions": [],

                "endpoint": {
                                 "url": {
                                     "ref": "Endpoint"
                                 },
                                 "properties": {},
                                 "headers": {}
                             },
                             "type": "endpoint"
                         }
                     ]
                 }
             ]
         },
         {
             "conditions": [],
             "type": "tree",
             "rules": [
                 {
                     "conditions": [
                         {
                             "fn": "isSet",

     "argv": [
                                 {
                                     "ref": "Region"
                                 }
                             ]
                         }
                     ],
                     "type": "tree",
                     "rules": [
                         {
                             "conditions": [
                                 {
                                     "fn": "aws.partition",
                                     "argv": [
                                         {
                                             "ref": "Region"
                                         }
                                     ],

 "assign": "PartitionResult"
                                 }
                             ],
                             "type": "tree",
                             "rules": [
```

```
                                    {
                               "conditions": [
                                  {
                                     "fn": "booleanEquals",
                                     "argv": [
                                        {
                                           "ref": "UseFIPS"
                                        },
                                        true
                                     ]
                                  },

             {
                                     "fn": "booleanEquals",
                                     "argv": [
                                        {
                                           "ref": "UseDualStack"
                                        },
                                        true
                                     ]
                                  }
                               ],
                               "type": "tree",
                               "rules": [
                                  {
                                     "conditions": [
                                        {

                               "fn": "booleanEquals",
                                           "argv": [
                                              true,
                                              {
                                                 "fn": "getAttr",
                                                 "argv": [
                                                    {
                                                       "ref": "PartitionResult"
                                                    },
                                                    "supportsFIPS"
                                                 ]

   }
                                           ]
                                        },
                                        {
                                           "fn": "booleanEquals",
                                           "argv": [
                                              true,
                                              {
```

```
                    "fn": "getAttr",
                    "argv": [
                      {
                        "ref": "PartitionResult"
                      },
                      "supportsDualStack"
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [],
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-linux-subscriptions-fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ]
                  }
                ]
              },
              {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or both",
                "type": "error"
              }
            ]
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
```

```
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,

{
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            }
        ],
        "type": "tree",

        "rules": [
            {
                "conditions": [],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                            "properties": {},
                            "headers": {}

                        },
                        "type": "endpoint"
                    }
                ]
            }
```

```
          ]
        },
        {
          "conditions": [],
          "error": "FIPS is enabled but this partition does not support FIPS",
          "type": "error"
        }
      ]

    },
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }
                      ],
                      "type": "tree",
                      "rules": [
                        {

                     "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
                                      "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                  ]
                                }
                              ]
                            }
                          ],
                          "type": "tree",
                          "rules": [
                            {
                              "conditions": [],
                              "type": "tree",
```

```
                              "rules": [
                                  {
                                      "conditions": [],

                                  "endpoint": {
                                          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                          "properties": {},
                                          "headers": {}
                                      },
                                      "type": "endpoint"
                                  }
                              ]
                          },

    {
                                  "conditions": [],
                                  "error": "DualStack is enabled but this partition does not support DualStack",
                                  "type": "error"
                              }
                          ]
                      },
                      {
                          "conditions": [],
                          "type": "tree",
                          "rules": [
                              {
                                  "conditions": [],
                                  "endpoint": {
                                      "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                      "properties": {},
                                      "headers": {}
                                  },
                                  "type": "endpoint"
                              }
                          ]
                      }
                  ]
              }
          ]
      },
      {
          "conditions": [],
          "error": "Invalid Configuration: Missing Region",
          "type": "error"
```

```
                }
              ]
            }
          ]
        },
        "smithy.rules#endpointTests": {
          "testCases": [
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
              "expect": {

                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
```

```
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {

      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
         "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
```

```
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint":
{

                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
             "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint":
```

```
{
                                 "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
                              }
                        },
                        "params": {
                           "Region": "us-gov-east-1",
                           "UseFIPS": false,
                           "UseDualStack": true
                        }
                     },
                     {
                        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                        "expect": {
                           "endpoint": {
                              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
                           }
                        },
                        "params": {
                           "Region": "us-gov-east-1",
                           "UseFIPS": false,
                            "UseDualStack": false
                        }
                     },
                     {
                        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                        "expect": {
                           "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                        },
                        "params": {
                           "Region": "us-iso-east-1",
                           "UseFIPS": true,
                           "UseDualStack": true
                        }
                     },
                     {
                        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                        "expect": {
                           "endpoint": {
                              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                           }
                        },
                        "params": {
                           "Region": "us-iso-east-1",
                           "UseFIPS": true,
                           "UseDualStack": false
                        }
                     },
                     {
```

```
            "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {

  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {

             "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
```

```
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled but this partition does not support DualStack"
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",

    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://example.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",

        "expect": {
                    "endpoint": {
                        "url": "https://example.com"
```

```
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,

    "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "Missing region",
              "expect": {
                "error": "Invalid Configuration: Missing Region"
              }
            }

      }
        ],
        "version": "1.0"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
```

```
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled LinuxSubscriptionsDiscovery"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type":
"operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with
commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {

        "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by
 the name of\n      <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n      <p>The valid names for this filter
are:</p>\n      <ul>\n         <li>\n            <p>\n               <code>AmiID</code>\n            </p>\n
</li>\n         <li>\n            <p>\n               <code>InstanceID</code>\n            </p>\n         </li>\n
<li>\n            <p>\n               <code>AccountID</code>\n            </p>\n         </li>\n         <li>\n
<p>\n               <code>Status</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
<code>Region</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n         </li>\n         <li>\n
    <p>\n               <code>ProductCode</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
    <code>InstanceType</code>\n            </p>\n         </li>\n      </ul>\n      <p>The valid Operators for this
filter are:</p>\n      <ul>\n         <li>\n            <p>\n               <code>contains</code>\n            </p>\n
</li>\n         <li>\n            <p>\n               <code>equals</code>\n            </p>\n         </li>\n         <li>\n
```

```
            <p>\n                <code>Notequal</code>\n            </p>\n        </li>\n    </ul>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

          }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
          "type": "structure",
          "members": {
            "Instances": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
              "traits": {
                "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
```

```
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n      organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
            }

          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't
match a certain subscription's name.</p>\n         <p>The valid names for this filter are:</p>\n         <ul>\n
<li>\n               <p>\n                  <code>Subscription</code>\n               </p>\n            </li>\n         </ul>\n
<p>The valid Operators for this filter are:</p>\n
      <ul>\n            <li>\n               <p>\n                  <code>contains</code>\n               </p>\n            </li>\n
<li>\n               <p>\n                  <code>equals</code>\n               </p>\n            </li>\n            <li>\n               <p>\n
```

```
          <code>Notequal</code>\n            </p>\n        </li>\n        </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>",
          "smithy.api#length": {

            "min": 1,
              "max": 16384
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "Subscriptions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
        }
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
```

```
                {
                    "name": "EQUAL",
                    "value": "Equal",
                    "documentation": "Equal operator"
                },
                {
                    "name": "NOT_EQUAL",
                    "value": "NotEqual",
                    "documentation": "Not equal operator"
                },
                {
                    "name": "CONTAINS",
                    "value": "Contains",
                    "documentation": "Contains operator"
                }
            ],
            "smithy.api#length": {
                "min": 1,
                "max": 20
            }
        }
    },

    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
```

```
      "traits": {
        "smithy.api#enum": [

  {
            "name": "InProgress",
            "value": "InProgress",
            "documentation": "InProgress status"
          },
          {
            "name": "Completed",
            "value": "Completed",
            "documentation": "Completed status"
          },
          {
            "name": "Successful",
            "value": "Successful",
            "documentation": "Successful status"
          },
          {
            "name": "Failed",
            "value": "Failed",
            "documentation": "Failed status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String",
        "traits":
{
          "smithy.api#length": {
            "min": 1,
            "max": 100
          }
        }
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
```

```
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription.</p>"

          }
          },
          "Type": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
          }
          },
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
```

```
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
      },

      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
            "smithy.api#required": {}
          }
```

```
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",

    "smithy.api#required": {}
                }
            },
            "AllowUpdate": {
                "target": "smithy.api#Boolean",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },

            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
```

```
            "traits": {
                "smithy.api#documentation":
"<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
 params = &ListLicensesInput{}
 }
```

```go
	result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*ListLicensesOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type ListLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//   - Beneficiary
	//   - ProductSKU
	//   - Fingerprint
	//
	//   - Status
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListLicensesOutput struct {

	// License details.
	Licenses []types.License

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}
```

```go
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if
 err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
```

```go
        err != nil {
            return err
        }
        if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
            return err
        }
        if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
            return err
        }
        if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
        err != nil {
            return err
        }
        if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
            return err
        }
        if err = addRequestIDRetrieverMiddleware(stack); err != nil {
            return err
        }
        if err = addResponseErrorMiddleware(stack); err != nil {
            return err
        }
        if err = addRequestResponseLogging(stack, options); err != nil {
            return err
        }
        if err = addDisableHTTPSMiddleware(stack, options); err != nil {
            return err
        }
        return nil
    }

    func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
        return &awsmiddleware.RegisterServiceMetadata{
            Region:       region,
            ServiceID:
                ServiceID,
            OperationName: "ListLicenses",
        }
    }
    // Code generated by smithy-go-codegen DO NOT EDIT.

    package grafana

    import (
        "context"
        "fmt"
        awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
        "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
```

```go
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 //
 This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```go
  noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
options, "DisassociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
```

```go
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
{
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "DisassociateLicense",
	}
}
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type
 CheckoutLicenseInput struct {

 // Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string
```

```go
// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource
Name (ARN) of the checkout license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string
```

```go
	// Signed token.
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil
	{
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
```

```go
    return err
  }
  if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
   return err
  }
  if err = addClientUserAgent(stack, options); err != nil {
   return err
  }
  if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
   return err
  }
  if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
   return err
  }
  if err = addSetLegacyContextSigningOptionsMiddleware(stack);
  err != nil {
   return err
  }
  if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
   return err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
   return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
   return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
   return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
   return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
   return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
   return err
  }
  return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
  return &awsmiddleware.RegisterServiceMetadata{
   Region:       region,
   ServiceID:    ServiceID,
```

```go
  OperationName: "CheckoutLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier
 that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see RFC 7518 Digital Signature with
 // RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
 //
```

```go
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
```

```go
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
```

```go
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return
err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:    region,
  ServiceID:   ServiceID,
```

```
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide
 to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string
```

```
// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity
*types.DatetimeRange

// Information about the license.
LicenseMetadata []types.Metadata


noSmithyDocumentSerde
```

```
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
		return fmt.Errorf("add
protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
```

```go
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
return err
}
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
```

```go
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName:
 "CreateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
```

```go
	//
	// This member is required.
	LicenseArn *string

	// License version.
	Version
	*string

	noSmithyDocumentSerde
}

type GetLicenseOutput struct {

	// License details.
	License *types.License

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return
	err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
```

```
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
if err = addOpGetLicenseValidationMiddleware(stack);
 err != nil {
 return err
 }
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
if err = addRequestResponseLogging(stack, options); err != nil {
```

```
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
```

```go
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
	if params == nil {
		params = &ListReceivedLicensesInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*ListReceivedLicensesOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type ListReceivedLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//
	//   - ProductSKU
	//   - Status
	//   - Fingerprint
	//   - IssuerName
	//   - Beneficiary
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
```

```
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
     return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
     return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack);
    err != nil {
     return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
     return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
     return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
    middleware.Before); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
    }
```

```
func newServiceMetadataMiddleware_opListReceivedLicenses(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:        region,
ServiceID:     ServiceID,
OperationName: "ListReceivedLicenses",
}
}
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

 4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct,
indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
    incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
```

```
                    }
                }
            },
            "traits": {
                "smithy.api#documentation":
"<p>Details about an Active Directory identity provider.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n        <note>\n          <p>Your estimated bill for charges on the number of users and related
costs will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
        monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
                "smithy.api#http": {
                    "code": 200,
```

```
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n      resource.</p>",
            "smithy.api#error": "server"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
```

```json
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
      "type": "operation",
```

```
"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}

},
  "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
```

```
                    "smithy.api#required": {}
                  }
                },
                "InstanceId": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                  }
                },
                "IdentityProvider": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                  "traits": {
                    "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                  }
                },
                "Domain": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The domain name of the user.</p>"
                  }
                }
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
            "type": "structure",
            "members": {
              "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                  "smithy.api#required": {}
                }
              }
            }
          },

    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
            "type": "structure",
            "members": {
              "Attribute": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
                }
```

```
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter name and value pair that is used
to return more specific results from a describe\n      operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n      attributes, or IDs.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n    settings such as the subnets to provision VPC endpoints.</p>",
              "smithy.api#required":
{}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
        "type": "structure",
        "members": {
          "InstanceId": {
```

```
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status
 of an EC2 instance resource.</p>",
                "smithy.api#required": {}
              }
            },
            "Products": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
                "smithy.api#required": {}
              }
            },
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
```

```
"members": {
    "Username": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits":
{
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
```

```
        },
        "DisassociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
          }
        }


    },
        "traits": {
          "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",

      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    }

  },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
     },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
```

```
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"

]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n    a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
                "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
                    "documentation": "The AWS region used to dispatch the request.",
                    "type": "String"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack
 endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                    "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }

            },
            "rules": [
                {
```

```
"conditions": [
  {
    "fn": "isSet",
    "argv": [
      {
        "ref": "Endpoint"
      }
    ]
  }
],
"type": "tree",
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },

          true
        ]
      }
    ],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseDualStack"
          },
          true
        ]
      }
    ],

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "endpoint": {
      "url": {
```

```
                            "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [

    {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "type": "tree",

"rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },
                                        true
                                    ]
```

```
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"

                        },
                        true
                      ]
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",

        "argv": [

                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                              ]
                            }
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {

          "fn": "getAttr",

                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                              ]
                            }
                          ]
                        }
```

```
                                    ],
                                    "type": "tree",
                                    "rules": [
                                      {

            "conditions": [],
                                        "endpoint": {
                                          "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                          "properties": {},
                                          "headers": {}
                                        },
                                        "type": "endpoint"
                                      }
                                    ]
                                  },
                                  {
                                    "conditions": [],
                                    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                                     "type": "error"
                                  }
                                ]
                              },
                              {
                                "conditions": [
                                  {
                                    "fn": "booleanEquals",
                                    "argv": [
                                      {
                                        "ref": "UseFIPS"
                                      },
                                      true
                                    ]
                                  }
                                ],
                                "type": "tree",

        "rules": [
                                  {
                                    "conditions": [
                                      {
                                        "fn": "booleanEquals",
                                        "argv": [
                                          true,
                                          {
                                            "fn": "getAttr",
                                            "argv": [
```

```
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"

                    ]
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}

        },

                        "type": "endpoint"
                }
            ]
        },
        {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"
        }
    ]
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
```

```
                                    {
                                      "fn": "booleanEquals",
                                      "argv": [
                                        true,
                                        {

                                          "fn": "getAttr",
                                            "argv": [
                                              {
                                                  "ref": "PartitionResult"
                                              },
                                              "supportsDualStack"
                                            ]
                                        }
                                      ]
                                    }
                                  ],
                                  "type": "tree",
                                  "rules": [
                                    {

                                      "conditions": [],
                                      "endpoint": {
                                          "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                          "properties": {},
                                          "headers": {}
                                      },
                                      "type": "endpoint"
                                    }
                                  ]
                                },
                                {
                                  "conditions": [],
                                  "error": "DualStack is enabled but this partition does
 not support DualStack",
                                  "type": "error"
                                }
                              ]
                            },
                            {
                              "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                              },
```

```
                    "type": "endpoint"
                  }
                ]
              }
            ]
          },
          {
            "conditions":
[],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
          }
        ]
      },
      "smithy.rules#endpointTests": {
        "testCases": [
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
```

```
                    }
                },

            "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with
 FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                    }
                },

        "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
```

```
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
              }
            },

      "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
```

```
                    "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
                }
            },

    "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
```

```
                {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                    "expect":
{
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                        }
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                        "error": "DualStack is enabled but this partition does not support DualStack"
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": false,
                        "UseDualStack":
true
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                        }
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": false,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                        "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
                    },
                    "params": {
                        "Region": "us-isob-east-1",
                        "UseFIPS": true,
```

```
                                    "UseDualStack": true
                                }
                            },
                            {
                                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                                "expect": {
                                    "endpoint": {
                                        "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                                    }
                                },
                                "params": {
                                    "Region": "us-isob-east-1",
                                    "UseFIPS": true,
                                    "UseDualStack": false
                                }

                },
                            {
                                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
                                "expect": {
                                    "error": "DualStack is enabled but this partition does not support DualStack"
                                },
                                "params": {
                                    "Region": "us-isob-east-1",
                                    "UseFIPS": false,
                                    "UseDualStack": true
                                }
                            },
                            {
                                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                                "expect": {
                                    "endpoint": {
                                        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                                    }
                                },
                                 "params": {
                                    "Region": "us-isob-east-1",
                                    "UseFIPS": false,
                                    "UseDualStack": false
                                }
                            },
                            {
                                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                                "expect": {
                                    "endpoint": {
                                        "url": "https://example.com"
                                    }
                                },
```

```
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {

"documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
             "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "Missing region",
          "expect": {
```

```
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
        },
        "smithy.api#paginated":
```

```
{
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
              "smithy.api#required": {}
          }
        },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
```

      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

```
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type":
"structure",
        "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListProductSubscriptions"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "ProductUserSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {

  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n    more sets of key-value pairs that you specify.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },

    "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
```

```
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
      },
      "smithy.api#paginated":
{
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
```

```
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
              }
```

```
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {

      "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }

      },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for a user.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
          }
        },
```

```
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
          }
        },
        "SubscriptionEndDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
```

```
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/RegisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n     subnets to provision VPC endpoints.</p>"
            }
          }
```

```
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 404
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
      "type": "string",

  "traits": {
        "smithy.api#length": {
          "min": 5,
          "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
```

```
              "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
              "smithy.api#error": "client"
          }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
          "type": "structure",
          "members": {
              "Subnets": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                  "traits": {
                      "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
                      "smithy.api#length": {
                          "min": 1
                      },
                      "smithy.api#required": {}
                  }
              },
              "SecurityGroupId": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                  "traits": {
                      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n      your VPC and the VPC endpoint for activation servers.</p>",
                      "smithy.api#required": {}
                  }
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n      subnets to provision VPC endpoints, and the security group ID that
 is associated with the VPC\n      endpoints. The security group should permit inbound TCP port 1688
communication from resources\n      in the VPC.</p>"
          }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
          "type": "operation",
          "input": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
          },
          "output": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
          },
          "errors": [
              {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
              },
              {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related costs
will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
 For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n        monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/StartProductSubscription"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
```

```
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

    },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
```

```
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required":
{}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
```

```
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation":
"<p>Metadata that describes the start product subscription operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {

          "target": "smithy.api#String"
        }
      },
      "traits": {
```

        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity\n   provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#required": {}
          }
        },
        "Product": {

```
                    "target": "smithy.api#String",
                    "traits": {
                      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                      "smithy.api#required": {}
                    }
                  }
                },
                "UpdateSettings": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                  "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n     update any combination of settings in a single operation such as the:</p>\n
<ul>\n          <li>\n                <p>Subnets which you want to add to provision VPC endpoints.</p>\n          </li>\n
        <li>\n                <p>Subnets which you want to remove the VPC endpoints from.</p>\n          </li>\n
<li>\n                <p>Security group ID which permits traffic to the VPC endpoints.</p>\n          </li>\n        </ul>",
                    "smithy.api#required": {}
                  }
                }
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
              "type": "structure",
              "members": {
                "IdentityProviderSummary": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                  "traits": {
                    "smithy.api#required": {}
                  }
                }
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
              "type": "structure",
              "members": {
                "AddSubnets": {

                  "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                  "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n     require connectivity to activation servers.</p>",
                    "smithy.api#required": {}
                  }
                },
                "RemoveSubnets": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                  "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                  }
```

```
              },
              "SecurityGroupId": {
                 "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                 "traits": {
                    "smithy.api#documentation": "<p>A security group ID
that allows inbound TCP port 1688 communication between resources in\n     your VPC and the VPC endpoints for
activation servers.</p>"
                 }
              }
           },
           "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as\n     the subnets to provision VPC endpoints.</p>"
           }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
           "type": "structure",
           "members": {
              "message": {
                 "target": "smithy.api#String"
              }
           },
           "traits": {
              "smithy.api#documentation": "<p>A parameter is not valid.</p>",
              "smithy.api#error": "client"
           }
        }
     }
}
{
   "smithy": "2.0",
   "metadata": {
      "suppressions": [
         {
            "id": "HttpMethodSemantics",
            "namespace": "*"
         },
         {
            "id": "HttpResponseCodeSemantics",
            "namespace": "*"
         },
         {
            "id": "PaginatedTrait",
            "namespace": "*"
         },
         {
            "id": "HttpHeaderTrait",
            "namespace": "*"
```

```
      },
      {
        "id": "HttpUriConflict",
        "namespace": "*"
      },
      {
        "id": "Service",
        "namespace": "*"
      }
    ]
  },
  "shapes": {
    "com.amazonaws.licensemanager#AWSLicenseManager": {
      "type": "service",
      "version": "2018-08-01",
      "operations": [
        {
          "target": "com.amazonaws.licensemanager#AcceptGrant"
        },


        {

          "target": "com.amazonaws.licensemanager#CheckInLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CheckoutLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateGrant"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateGrantVersion"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
        },
        {
```

```
      "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateToken"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
      "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    }
  },
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
    {
      "target": "com.amazonaws.licensemanager#GetGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseUsage"
    },
    {
      "target": "com.amazonaws.licensemanager#GetServiceSettings"
    },
```

```
{
    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
},
{
    "target": "com.amazonaws.licensemanager#ListDistributedGrants"
},
{
    "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenses"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
},
{

    "target": "com.amazonaws.licensemanager#ListLicenseVersions"
},
{
    "target": "com.amazonaws.licensemanager#ListReceivedGrants"
},
{
    "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
},
{
    "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
},
{
    "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
},
{
    "target": "com.amazonaws.licensemanager#ListResourceInventory"
},
{
    "target": "com.amazonaws.licensemanager#ListTagsForResource"
},
{
    "target": "com.amazonaws.licensemanager#ListTokens"
```

```json
      },
      {
        "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
      },
      {
        "target": "com.amazonaws.licensemanager#RejectGrant"
      },
      {
        "target": "com.amazonaws.licensemanager#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UntagResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager",
        "arnNamespace": "license-manager",
        "cloudFormationName": "LicenseManager",
        "cloudTrailEventSource": "licensemanager.amazonaws.com",
        "endpointPrefix": "license-manager"
      },
      "aws.auth#sigv4": {
        "name": "license-manager"
      },
      "aws.protocols#awsJson1_1": {},
      "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n        Amazon Web Services accounts and on-premises servers.</p>",
      "smithy.api#title": "AWS License Manager",
      "smithy.api#xmlNamespace": {
        "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
      },
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
```

```
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "String"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "Boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]

        }
      ],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {
```

```
                                        "fn": "booleanEquals",
                                        "argv": [
                                          {
                                            "ref": "UseFIPS"
                                          },
                                          true
                                        ]
                                      }
                                    ],
                                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                                    "type": "error"
                                  },
                                  {

"conditions": [
                                      {
                                        "fn": "booleanEquals",
                                        "argv": [
                                          {
                                            "ref": "UseDualStack"
                                          },
                                          true
                                        ]
                                      }
                                    ],
                                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                                    "type": "error"
                                  },
                                  {
                                    "conditions": [],
                                    "endpoint": {
                                      "url": {
                                        "ref": "Endpoint"
                                      },
                                      "properties": {},
                                      "headers": {}
                                    },
                                    "type": "endpoint"
                                  }
                                ]
                              },
                              {
                                "conditions": [
                                  {
                                    "fn": "isSet",
                                    "argv": [
                                      {
                                        "ref": "Region"
```

```
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "aws.partition",
                      "argv": [
                        {
                          "ref": "Region"
                        }
                      ],
                      "assign": "PartitionResult"
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref":
"UseFIPS"
                            },
                            true
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }
                      ],
                      "type": "tree",
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
```

```
                                            "argv": [
                                              true,
                                              {
                                                "fn": "getAttr",
                                                "argv": [
                                                  {
                                                    "ref": "PartitionResult"
                                                  },
                                                  "supportsFIPS"
                                                ]

                                }
                                  ]
                                },
                                {
                                  "fn": "booleanEquals",
                                  "argv": [
                                    true,
                                    {
                                      "fn": "getAttr",
                                      "argv": [
                                        {
                                          "ref": "PartitionResult"
                                        },

    "supportsDualStack"

                                        ]
                                      }
                                    ]
                                  }
                                ],
                                "type": "tree",
                                "rules": [
                                  {
                                    "conditions": [],
                                    "endpoint": {
                                      "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                      "properties": {},
                                      "headers": {}

                            },
                                    "type": "endpoint"
                                  }
                                ]
                              },
                              {
                                "conditions": [],
```

```
                              "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                              "type": "error"
                           }
                        ]
                    },
                    {
                       "conditions": [
                          {
                             "fn": "booleanEquals",

                       "argv": [
                                {
                                   "ref": "UseFIPS"
                                },
                                true
                             ]
                          }
                       ],
                       "type": "tree",
                       "rules": [
                          {
                             "conditions": [
                                {
                                   "fn": "booleanEquals",
                                   "argv": [
                                      true,

                    {
                                         "fn": "getAttr",
                                         "argv": [
                                            {
                                               "ref": "PartitionResult"
                                            },
                                            "supportsFIPS"
                                         ]
                                      }
                                   ]
                                }
                             ],
                             "type": "tree",
                             "rules": [

                    {
                                   "conditions": [],
                                   "endpoint": {
                                      "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                                      "properties": {},
```

```
                              "headers": {}
                          },
                          "type": "endpoint"
                      }
                  ]
              },
              {
                  "conditions": [],
                  "error": "FIPS is enabled but this
partition does not support FIPS",
                  "type": "error"
              }
          ]
      },
      {
          "conditions": [
              {
                  "fn": "booleanEquals",
                  "argv": [
                      {
                          "ref": "UseDualStack"
                      },
                      true
                  ]
              }
          ],
          "type": "tree",
          "rules": [
              {
                  "conditions": [
                      {
                          "fn": "booleanEquals",
                          "argv": [
                              true,
                              {
                                  "fn": "getAttr",
                                  "argv": [
                                      {
                                          "ref": "PartitionResult"
                                      },

"supportsDualStack"
                                  ]
                              }
                          ]
                      }
                  ],
                  "type": "tree",
```

```
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}

                                        },
                                        "type": "endpoint"
                                    }
                                ]
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ]
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                             "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
  ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint":
```

```
{
                                "url": "https://license-manager.af-south-1.amazonaws.com"
                            }
                        },
                        "params": {
                            "Region": "af-south-1",
                            "UseFIPS": false,
                            "UseDualStack": false
                        }
                    },
                    {
                        "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager.ap-east-1.amazonaws.com"
                            }
                        },
                        "params": {
                            "Region": "ap-east-1",
                            "UseFIPS": false,
                            "UseDualStack": false
                        }
                    },
                    {
                        "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager.ap-northeast-1.amazonaws.com"
                            }
                        },
                        "params": {
                            "Region": "ap-northeast-1",
                            "UseFIPS": false,
                            "UseDualStack": false
                        }
                    },
                    {
                        "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager.ap-northeast-2.amazonaws.com"
                            }
                        },
                        "params": {
                            "Region": "ap-northeast-2",
                            "UseFIPS": false,
                            "UseDualStack": false
                        }
```

```
        },
        {
          "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-3.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }

      },
        {

          "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-1.amazonaws.com"

  }
          },
          "params": {
            "Region": "ap-southeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-2.amazonaws.com"
            }
```

```
      },
      "params": {
        "Region": "ap-southeast-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-southeast-3.amazonaws.com"
        }
      },
      "params": {
        "Region": "ap-southeast-3",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ca-central-1.amazonaws.com"
        }
      },

     "params": {
        "Region": "ca-central-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.eu-central-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "eu-central-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
```

```
          {
            "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.eu-north-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "eu-north-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.eu-south-1.amazonaws.com"
              }
            },
            "params": {
              "Region":
"eu-south-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.eu-west-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "eu-west-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
            "expect": {

              "endpoint": {
                "url": "https://license-manager.eu-west-2.amazonaws.com"
              }
```

```
        },
        "params": {
          "Region": "eu-west-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.eu-west-3.amazonaws.com"
          }
        },
        "params": {
          "Region": "eu-west-3",
          "UseFIPS": false,

"UseDualStack": false
        }
      },
      {
        "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.me-south-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "me-south-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.sa-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "sa-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
```

```
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }

      },
            {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
      },
          {
          "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-east-2.amazonaws.com"
            }

        },
          "params": {
            "Region": "us-east-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
      },
          {
          "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-east-2.amazonaws.com"
            }
          },
          "params": {
```

```
                "Region": "us-east-2",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region
us-west-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.us-west-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-west-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-fips.us-west-1.amazonaws.com"
                }
            },
            "params": {

        "Region": "us-west-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.us-west-2.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-west-2",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",
```

```
"expect": {
    "endpoint": {
        "url": "https://license-manager-fips.us-west-2.amazonaws.com"
    }
},
"params": {
    "Region": "us-west-2",
    "UseFIPS": true,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,

        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
```

```
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                 "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-northwest-1",
              "UseFIPS": false,
              "UseDualStack": false
            }

}
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                 "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                 "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
```

```
          "endpoint": {
            "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },

  {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
          }
         },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-west-1",
          "UseFIPS": false,
          "UseDualStack": false
```

```
              }
            },
            {
              "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-gov-west-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.us-gov-east-1.api.aws"
                }
              },

      "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.us-gov-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
              "expect": {
                "endpoint": {
```

```
                    "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
                  }
                },
                "params": {
                  "Region": "us-iso-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {
                  "Region": "us-iso-east-1",
                  "UseFIPS": true,
                   "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
                  }
                },
                "params": {
                  "Region": "us-iso-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "error": "DualStack is enabled
but this partition does not support DualStack"
                },
                "params": {
                  "Region": "us-iso-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
```

```
          "endpoint": {
            "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
          }
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },

    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
          }
        },

"params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
```

"documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

"expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,

"UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {

```
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "Missing region",
                "expect": {
                    "error": "Invalid Configuration: Missing Region"
                }
            }
        ],
        "version": "1.0"
    }
}
},
"com.amazonaws.licensemanager#AcceptGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#AcceptGrantRequest"
    }
},
    "output": {
        "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
```

```
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant
ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
```

```
        "com.amazonaws.licensemanager#AccessDeniedException": {
            "type": "structure",
            "members": {
                "Message": {
                    "target": "com.amazonaws.licensemanager#Message"
                }
            },
            "traits": {
                "aws.protocols#awsQueryError": {

"code": "ServiceAccessDenied",
                    "httpResponseCode": 401
                },
                "smithy.api#documentation": "<p>Access to resource denied.</p>",
                "smithy.api#error": "client",
                "smithy.api#httpError": 401
            }
        },
        "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
            "type": "enum",
            "members": {
                "DISTRIBUTED_GRANTS_ONLY": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
                    }
                },
                "ALL_GRANTS_PERMITTED_BY_ISSUER": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#AllowedOperation": {

"type": "enum",
            "members": {
                "CREATE_GRANT": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "CreateGrant"
                    }
                },
                "CHECKOUT_LICENSE": {
                    "target": "smithy.api#Unit",
                    "traits": {
```

```
            "smithy.api#enumValue": "CheckoutLicense"
          }
        },
        "CHECKOUT_BORROW_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutBorrowLicense"
          }
        },
        "CHECK_IN_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckInLicense"
          }
        },
        "EXTEND_CONSUMPTION_LICENSE":
{

          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ExtendConsumptionLicense"
          }
        },
        "LIST_PURCHASED_LICENSES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ListPurchasedLicenses"
          }
        },
        "CREATE_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateToken"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#AllowedOperationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#AllowedOperation"
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max":
7
        }
      }
```

```
    },
    "com.amazonaws.licensemanager#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 2048
        },
        "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
      }
    },
    "com.amazonaws.licensemanager#ArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
      }
    },
    "com.amazonaws.licensemanager#AuthorizationException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError":
{
          "code": "AuthorizationFailure",
          "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to perform the action. Check the IAM\n        policy associated with this account.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 403
      }
    },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
      "type": "structure",
      "members": {
        "LastRunTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
          }
        }
      },
      "traits": {
```

```
          "smithy.api#documentation": "<p>Describes automated discovery.</p>"
        }
      },

  "com.amazonaws.licensemanager#Boolean": {
      "type": "boolean",
      "traits": {
        "smithy.api#default": false
      }
    },
    "com.amazonaws.licensemanager#BorrowConfiguration": {
      "type": "structure",
      "members": {
        "AllowEarlyCheckIn": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxTimeToLiveInMinutes": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
            "smithy.api#required": {}
          }
        }

  },
      "traits": {
        "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#BoxBoolean": {
      "type": "boolean"
    },
    "com.amazonaws.licensemanager#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanager#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#CheckInLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
      },
      "output": {
```

```
            "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"

        },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
        }
      },
      "com.amazonaws.licensemanager#CheckInLicenseRequest":
  {
        "type": "structure",
        "members": {
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckInLicenseResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
          "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
```

```
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
        "type": "structure",
        "members":
  {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
              "smithy.api#required": {}
            }
          },
          "DigitalSignatureMethod": {
            "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
            "traits": {
              "smithy.api#documentation": "<p>Digital signature method. The possible
 value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
              "smithy.api#required": {}
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
```

```
              }
            },
            "CheckoutMetadata": {
              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                "smithy.api#documentation": "<p>Information about constraints.</p>"
              }
            },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientToken",
              "traits": {
                "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
              }
            },
            "LicenseConsumptionToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License consumption token.</p>"
              }
            },
            "EntitlementsAllowed":
{
              "target": "com.amazonaws.licensemanager#EntitlementDataList",
              "traits": {
                "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
              }
            },
            "NodeId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Node ID.</p>"
              }
```

```
            },
            "SignedToken": {
              "target": "com.amazonaws.licensemanager#SignedToken",
              "traits": {
                "smithy.api#documentation": "<p>Signed token.</p>"
              }
            },
            "IssuedAt": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
              }
            },
            "Expiration": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
              }
            },
            "CheckoutMetadata": {
              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                "smithy.api#documentation": "<p>Information about constraints.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target":
"com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n            <p>If the
account that created the license is the same that is performing the check out, you must\n            specify the account
as the beneficiary.</p>\n        </note>"
    }
  },
  "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
    "type": "structure",
    "members": {
      "ProductSKU": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product
SKU.</p>",
          "smithy.api#required": {}
        }
      },
      "CheckoutType": {
        "target": "com.amazonaws.licensemanager#CheckoutType",
        "traits": {
          "smithy.api#documentation": "<p>Checkout type.</p>",
          "smithy.api#required": {}
        }
      },
```

```
        "KeyFingerprint": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        }
      },
      "traits": {

    "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
      "type": "structure",
      "members": {
        "CheckoutType": {
          "target": "com.amazonaws.licensemanager#CheckoutType",
          "traits": {
            "smithy.api#documentation": "<p>Checkout type.</p>"
```

```
        }
      },
      "LicenseConsumptionToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License consumption token.</p>"
        }
      },
      "EntitlementsAllowed": {
        "target": "com.amazonaws.licensemanager#EntitlementDataList",
        "traits": {
          "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
        }
      },
       "SignedToken": {
        "target": "com.amazonaws.licensemanager#SignedToken",
        "traits": {
          "smithy.api#documentation": "<p>Signed token.</p>"
        }
      },
      "NodeId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Node ID.</p>"
        }
      },
      "IssuedAt": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
        }
      },
      "Expiration": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
        }
      },
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
```

```
            }
        },
        "com.amazonaws.licensemanager#CheckoutType": {
            "type": "enum",
            "members": {
                "PROVISIONAL": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "PROVISIONAL"
                    }
                },
                "PERPETUAL": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "PERPETUAL"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#ClientRequestToken": {
            "type": "string",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 36
                }
            }
        },
        "com.amazonaws.licensemanager#ClientToken": {
            "type": "string",
            "traits": {
                "smithy.api#length": {
                    "min": 0,
                    "max": 2048
                },
                "smithy.api#pattern": "^\\S+$"
            }
        },
        "com.amazonaws.licensemanager#ConflictException": {
            "type": "structure",
            "members": {
                "Message": {
                    "target": "com.amazonaws.licensemanager#Message"
                }
            },
            "traits": {
                "aws.protocols#awsQueryError": {
                    "code": "ConflictException",
```

```
                  "httpResponseCode":
    409
              },
              "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
    again.</p>",
              "smithy.api#error": "client",
              "smithy.api#httpError": 409
          }
      },
      "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
          "type": "structure",
          "members": {
              "ResourceType": {
                  "target": "com.amazonaws.licensemanager#ResourceType",
                  "traits": {
                      "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
                  }
              },
              "ConsumedLicenses": {
                  "target": "com.amazonaws.licensemanager#BoxLong",
                  "traits": {
                      "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
                  }
              }
          },

      "traits": {
              "smithy.api#documentation": "<p>Details about license consumption.</p>"
          }
      },
      "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
          "type": "list",
          "member": {
              "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
          }
      },
      "com.amazonaws.licensemanager#ConsumptionConfiguration": {
          "type": "structure",
          "members": {
              "RenewType": {
                  "target": "com.amazonaws.licensemanager#RenewType",
                  "traits": {
                      "smithy.api#documentation": "<p>Renewal frequency.</p>"
                  }
              },
              "ProvisionalConfiguration": {
                  "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
                  "traits": {
```

```
              "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
            }
        },
        "BorrowConfiguration": {
          "target": "com.amazonaws.licensemanager#BorrowConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target":
"com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
```

license\n   entitlements with a specific Amazon Web Services account, an organization, or an\n organizational unit (OU). For more
 information, see Granted licenses in License Manager in the <i>License Manager User Guide</i>.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateGrantRequest": {
        "type": "structure",
        "members": {
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant name.</p>",
                    "smithy.api#required": {}

    }
        }
    },
        "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                "smithy.api#required": {}
            }
        },
        "Principals": {
            "target": "com.amazonaws.licensemanager#PrincipalArnList",
            "traits": {
                "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an Amazon Resource Name\n   (ARN):</p>\n   \n   \n   <p>An Amazon Web Services account, which includes only the account specified.</p>\n   \n   \n   \n   \n   <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n   \n   \n \n
        \n   <p>An organization, which will include all accounts across your organization.</p>\n \n   ",
                "smithy.api#required": {}
            }
        },
        "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

```
                    "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "AllowedOperations": {
                "target": "com.amazonaws.licensemanager#AllowedOperationList",
                "traits": {
                    "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

    "com.amazonaws.licensemanager#CreateGrantResponse": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant version.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateGrantVersion":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
```

```
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
        "type": "structure",
        "members": {
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
                }
            },
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
```

```
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant name.</p>"
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "StatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Grant status reason.</p>"
          }
        },
        "SourceVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the grant.</p>"
          }
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
            "smithy.api#documentation": "<p>The options specified for the grant.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
      "type": "structure",
      "members": {
```

```
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>New version of the grant.</p>"
          }
        }
      },
      "traits":
{
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of
licenses purchased and used.</p>"
      }
```

        },
      "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the license configuration.</p>"
            }
          },
          "LicenseCountingType": {
            "target": "com.amazonaws.licensemanager#LicenseCountingType",
            "traits": {
              "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
            }
          },
          "LicenseCountHardLimit": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n   blocks the launch of new instances.</p>"
            }
          },
          "LicenseRules": {
            "target":
 "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n   vary by dimension, as follows.</p>\n
\n   \n   <p>\n   <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
      <code>licenseAffinityToHost</code> |\n   <code>maximumCores</code> |
<code>minimumCores</code>\n   </p>\n   \n   \n   <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n   <code>maximumCores</code> |
<code>minimumCores</code> |\n   <code>maximumSockets</code> | <code>minimumSockets</code> |\n

<code>maximumVcpus</code> | <code>minimumVcpus</code>\n   </p>\n   \n
\n
  <p>\n   <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n   <code>maximumSockets</code> |
<code>minimumSockets</code>\n   </p>\n   \n   \n   <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n   <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n   </p>\n   \n   \n   <p>The unit for
<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for
<code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n
<code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n
<code>True</code> and <code>False</code>.</p>"

              }
            },

    "Tags": {
              "target": "com.amazonaws.licensemanager#TagList",
              "traits": {
                "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
              }
            },
            "DisassociateWhenNotFound": {
              "target": "com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
              }
            },
            "ProductInformationList": {
              "target": "com.amazonaws.licensemanager#ProductInformationList",
              "traits": {
                "smithy.api#documentation": "<p>Product information.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
              }

```
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {

            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
              "smithy.api#required": {}
```

```
            }
        },
        "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
                "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n        For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
                "smithy.api#required": {}
            }
        },
        "DestinationLicenseContext":
{
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
                "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
    "type": "structure",
    "members": {
        "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

    "traits": {
                "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
    },
```

```
"output": {
  "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanager#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanager#AuthorizationException"
  },
  {
    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
  },
  {
    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
  },
  {
    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
  },
  {
    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanager#ServerInternalException"
  },
  {
    "target": "com.amazonaws.licensemanager#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a report generator.</p>"
}
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
  "type": "structure",
  "members": {
    "ReportGeneratorName":
    {
      "target": "com.amazonaws.licensemanager#ReportGeneratorName",
      "traits": {
        "smithy.api#documentation": "<p>Name of the report generator.</p>",
        "smithy.api#required": {}
      }
    },
    "Type": {
      "target": "com.amazonaws.licensemanager#ReportTypeList",
      "traits": {
        "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
```

generated:</p>\n   \n   \n   <p>License configuration report - Reports the number and details of consumed licenses for a license configuration.</p>\n   \n   \n   <p>Resource report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n   \n
",

                "smithy.api#required": {}

            }
        },
        "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
                "smithy.api#documentation": "<p>Defines the type of license configuration the report generator tracks.</p>",
                "smithy.api#required": {}
            }
        },
        "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits": {
                "smithy.api#documentation": "<p>Frequency by which reports are generated. Reports can be generated daily, monthly, or weekly.</p>",
                "smithy.api#required": {}
            }
        },
        "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientRequestToken",
            "traits": {
                "smithy.api#documentation": "<p>Unique,
 case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                "smithy.api#required": {}
            }
        },
        "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
                "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }

```
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn":
  {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product
name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>",
            "smithy.api#required": {}
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#Issuer",
          "traits": {
            "smithy.api#documentation": "<p>License issuer.</p>",
            "smithy.api#required": {}
          }
```

```
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region for the license.</p>",
            "smithy.api#required": {}

  }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>",
            "smithy.api#required": {}
          }
        },
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n         running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n         usage.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
```

```
                "traits": {
                    "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseResponse": {
        "type": "structure",
        "members": {
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#LicenseStatus",
                "traits": {
                    "smithy.api#documentation": "<p>License status.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License version.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersion": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target":
    "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                "smithy.api#required": {}
              }
            },
            "LicenseName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License name.</p>",
                "smithy.api#required": {}
              }
            },
            "ProductName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {

```
          "smithy.api#documentation": "<p>Product name.</p>",
          "smithy.api#required": {}
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#Issuer",
        "traits":

{

          "smithy.api#documentation": "<p>License issuer.</p>",
          "smithy.api#required": {}
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the license.</p>",
          "smithy.api#required": {}
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
           "smithy.api#documentation": "<p>Information about the license.</p>"
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>",
          "smithy.api#required": {}
        }
      },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
          "smithy.api#required": {}
        }
```

                },

    "Status": {
                "target": "com.amazonaws.licensemanager#LicenseStatus",
                "traits": {
                  "smithy.api#documentation": "<p>License status.</p>",
                  "smithy.api#required": {}
                }
              },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientToken",
              "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            },
            "SourceVersion": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Current version of the license.</p>"
              }
            }
          },
        "traits": {

 "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New version of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"

```
          }
        }
      },
      "traits": {
        "smithy.api#output":
{}
      }
    },
    "com.amazonaws.licensemanager#CreateToken": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateTokenRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateTokenResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },

        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
      }
```

```
      },
    "com.amazonaws.licensemanager#CreateTokenRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits":
{
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
              "smithy.api#required": {}
          }
        },
        "RoleArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n        License Manager does not check whether the roles are in use.</p>"
          }
        },
        "ExpirationInDays": {
          "target": "com.amazonaws.licensemanager#Integer",
          "traits": {
              "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
              "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n        to the amr claim of the JWT token.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
              "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
              "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateTokenResponse": {
      "type": "structure",
```

```
    "members": {
      "TokenId": {
        "target": "com.amazonaws.licensemanager#String",


       "traits": {
          "smithy.api#documentation": "<p>Token ID.</p>"
        }
      },
      "TokenType": {
        "target": "com.amazonaws.licensemanager#TokenType",
        "traits": {
          "smithy.api#documentation": "<p>Token type.</p>"
        }
      },
      "Token": {
        "target": "com.amazonaws.licensemanager#TokenString",
        "traits": {
          "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#DateTime": {
    "type": "timestamp"
  },
  "com.amazonaws.licensemanager#DatetimeRange": {
    "type": "structure",
    "members": {
      "Begin": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Start of the time range.</p>",
          "smithy.api#required": {}
        }
      },
      "End": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>End of the time range.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
    }
```

```json
        },
        "com.amazonaws.licensemanager#DeleteGrant": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
            }
        },
        "com.amazonaws.licensemanager#DeleteGrantRequest": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                        "smithy.api#required": {}
                    }
                },
                "StatusReason": {
```

```
                    "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                    "traits": {
                        "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Current version of the
grant.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteGrantResponse": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant ARN.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation":
"<p>Grant version.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteLicense": {
            "type": "operation",
            "input": {
```

```
          "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"

},
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
        },

    "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
        "type": "structure",
        "members":
{
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the license configuration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
        },
        "output": {
```

          "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n   <p>This action
deletes the report generator, which stops it from generating future reports.\n   The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

```json
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "SourceVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the license.</p>",
            "smithy.api#required":
{}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseResponse": {
      "type": "structure",
      "members": {
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "DeletionDate": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
          }
        }
```

```
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteToken": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },

          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteTokenRequest": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
                        "smithy.api#documentation": "<p>Token ID.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteTokenResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}

            }
        },
        "com.amazonaws.licensemanager#DigitalSignatureMethod": {
            "type": "enum",
            "members": {
                "JWT_PS384": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "JWT_PS384"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#Entitlement": {
            "type": "structure",
            "members": {
                "Name": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Entitlement name.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Value": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
                    }
                },
                "MaxCount": {
                    "target": "com.amazonaws.licensemanager#Long",
                    "traits": {
```

```
              "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
            }
        },
        "Overage": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement unit.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowCheckIn": {
          "target":
"com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementData": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement data name.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement
data value.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>Entitlement data unit.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
        }
    },
    "com.amazonaws.licensemanager#EntitlementDataList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#EntitlementData"
        }
    },
    "com.amazonaws.licensemanager#EntitlementDataUnit": {
        "type": "enum",
        "members": {
            "COUNT": {
                "target": "smithy.api#Unit",
                "traits": {

"smithy.api#enumValue": "Count"
                }
            },
            "NONE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "None"
                }
            },
            "SECONDS": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Seconds"
                }
            },
            "MICROSECONDS": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Microseconds"
                }
            },
            "MILLISECONDS": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Milliseconds"
                }
```

```
        },
        "BYTES": {
          "target": "smithy.api#Unit",
           "traits": {
            "smithy.api#enumValue": "Bytes"
          }
        },
        "KILOBYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes"
          }
        },
        "MEGABYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes"
          }
        },
        "GIGABYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes"
          }
        },
        "TERABYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabytes"
          }
        },
        "BITS":
{
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits"
          }
        },
        "KILOBITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits"
          }
        },
        "MEGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits"
```

```
          }
        },
        "GIGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits"
          }
        },
        "TERABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits"

}
        },
        "PERCENT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Percent"
          }
        },
        "BYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bytes/Second"
          }
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
           "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
```

```
            "smithy.api#enumValue": "Terabytes/Second"
          }
        },
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
          }
        },
        "TERABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits/Second"
          }
        },
        "COUNT_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count/Second"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Entitlement"
      }
    },
```

```
"com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
},
    "com.amazonaws.licensemanager#EntitlementUnit": {
      "type": "enum",
      "members": {
        "COUNT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count"
          }
        },
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "SECONDS":
{
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Seconds"
          }
        },
        "MICROSECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Microseconds"
          }
        },
        "MILLISECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Milliseconds"
          }
        },
```

```
"BYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Bytes"
  }
},
"KILOBYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Kilobytes"
  }
},
"MEGABYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Megabytes"
  }
},
"GIGABYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Gigabytes"
  }
},
"TERABYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Terabytes"
  }
},
"BITS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Bits"
  }
},
"KILOBITS": {
  "target": "smithy.api#Unit",
  "traits": {

    "smithy.api#enumValue": "Kilobits"
  }
},
"MEGABITS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Megabits"
  }
```

```
        },
        "GIGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits"
          }
        },
        "TERABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits"
          }
        },
        "PERCENT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Percent"
          }
        },
        "BYTES_PER_SECOND": {
          "target":
"smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bytes/Second"
          }
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
```

```
"smithy.api#enumValue": "Terabytes/Second"
            }
        },
        "BITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Bits/Second"
            }
        },
        "KILOBITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Kilobits/Second"
            }
        },
        "MEGABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Megabits/Second"
            }
        },
        "GIGABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Gigabits/Second"
            }
        },

        "TERABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Terabits/Second"
            }
        },
        "COUNT_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Count/Second"
            }
        }
    }
},
"com.amazonaws.licensemanager#EntitlementUsage": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```json
            "smithy.api#documentation": "<p>Entitlement usage name.</p>",
            "smithy.api#required": {}
          }
        },
        "ConsumedValue": {
          "target": "com.amazonaws.licensemanager#String",

          "traits": {
            "smithy.api#documentation": "<p>Resource usage consumed.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList":
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementUsage"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
         },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>",
            "smithy.api#required": {}

    }
        },
        "DryRun": {
          "target": "com.amazonaws.licensemanager#Boolean",
          "traits": {
            "smithy.api#default": false,
            "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
```

```
"com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
    "type": "structure",
    "members": {
        "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>License consumption token.</p>"
            }
        },

        "Expiration": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#FailedDependencyException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        },
        "ErrorCode": {
            "target": "com.amazonaws.licensemanager#String"
        }
    },
    "traits": {
        "aws.protocols#awsQueryError": {
            "code": "FailedDependency",
            "httpResponseCode": 424
        },
        "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 424
    }
},
"com.amazonaws.licensemanager#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "com.amazonaws.licensemanager#FilterName",
            "traits": {
```

```
          "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanager#FilterValues",
        "traits": {
          "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n        describe operation. Filters can be used
to match a set of resources by specific criteria,\n        such as tags, attributes, or IDs.</p>"
    }
  },
  "com.amazonaws.licensemanager#FilterLimitExceededException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "FilterLimitExceeded",
        "httpResponseCode": 400
      },
      "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Filter"
    }
  },
  "com.amazonaws.licensemanager#FilterName": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValue": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValues": {
    "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanager#FilterValue",
          "traits": {
            "smithy.api#xmlName": "item"
          }
        }
      },
      "com.amazonaws.licensemanager#Filters": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Filter",
          "traits": {
            "smithy.api#xmlName": "item"
          }
        }
      },
      "com.amazonaws.licensemanager#GetAccessToken": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
tokens\n        are valid for one hour.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetAccessTokenRequest": {
```

```
          "type": "structure",
          "members": {
            "Token": {
              "target": "com.amazonaws.licensemanager#TokenString",
              "traits": {
                "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
                "smithy.api#required": {}
              }
            },
            "TokenProperties": {
              "target": "com.amazonaws.licensemanager#MaxSize3StringList",
              "traits": {
                "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
          "type": "structure",
          "members": {
            "AccessToken": {
              "target": "com.amazonaws.licensemanager#TokenString",
              "traits": {
                "smithy.api#documentation": "<p>Temporary access token.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#GetGrant": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#GetGrantRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#GetGrantResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
```

```
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"

          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetGrantResponse": {
        "type": "structure",
        "members": {
```

```
      "Grant": {
        "target": "com.amazonaws.licensemanager#Grant",
        "traits": {
          "smithy.api#documentation": "<p>Grant details.</p>"
        }
      }
    },
    "traits": {

"smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#GetLicense": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetLicenseRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetLicenseResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#GetLicenseConfiguration": {
    "type": "operation",
    "input": {
```

```
          "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits":
{
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
        "type": "structure",
        "members": {
          "LicenseConfigurationId": {
```

```
            "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
          }
        },
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules.</p>"
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",

          "traits": {
            "smithy.api#documentation": "<p>Number of available licenses.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
          }
        },
```

```
    "ConsumedLicenses": {
      "target": "com.amazonaws.licensemanager#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>License configuration status.</p>"
      }

},
    "OwnerAccountId": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
      }
    },
    "ConsumedLicenseSummaryList": {
      "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
      }
    },
    "ManagedResourceSummaryList": {
      "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanager#TagList",
       "traits": {
        "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
      }
    },
    "ProductInformationList": {
      "target": "com.amazonaws.licensemanager#ProductInformationList",
      "traits": {
        "smithy.api#documentation": "<p>Product information.</p>"
      }
    },
    "AutomatedDiscoveryInformation": {
      "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
      "traits": {
        "smithy.api#documentation": "<p>Automated discovery information.</p>"
      }
```

```
            },
            "DisassociateWhenNotFound": {
               "target": "com.amazonaws.licensemanager#BoxBoolean",
               "traits": {
                  "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTask": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
               "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {

"target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
               "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
         }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
         "type": "structure",
         "members": {
            "LicenseConversionTaskId": {
               "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
```

```
            "traits": {
              "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
```

```
                    "traits": {
                        "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
                    }
                },

    "StartTime": {
                    "target": "com.amazonaws.licensemanager#DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                    }
                },
                "LicenseConversionTime": {
                    "target": "com.amazonaws.licensemanager#DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                    }
                },
                "EndTime": {
                    "target": "com.amazonaws.licensemanager#DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
```

```
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
              "smithy.api#required":
{}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
      "type": "structure",
      "members": {
        "ReportGenerator": {
          "target": "com.amazonaws.licensemanager#ReportGenerator",
          "traits": {
            "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
          }
        }
      },
```

```
          "traits": {
            "smithy.api#output": {}
          }
        },
      "com.amazonaws.licensemanager#GetLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseResponse": {
        "type": "structure",
        "members": {
          "License": {
            "target": "com.amazonaws.licensemanager#License",
            "traits": {
              "smithy.api#documentation": "<p>License details.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },

    "com.amazonaws.licensemanager#GetLicenseUsage": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
```

          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {

      "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
          "type": "structure",
          "members": {
            "LicenseUsage": {
              "target":
"com.amazonaws.licensemanager#LicenseUsage",
              "traits": {

```
            "smithy.api#documentation": "<p>License usage details.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
```

discovery data,\n   and so on.</p>"

```
 }
        },
        "SnsTopicArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
          }
        },
        "OrganizationConfiguration": {
          "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n        cross-account discovery.</p>"
          }
        },
        "EnableCrossAccountsDiscovery": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
          }

      },
        "LicenseManagerResourceShareArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n        provides member accounts with access to this share.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Grant": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
```

```
          "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant name.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ParentArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Parent ARN.</p>",
                        "smithy.api#required": {}
                    }
                },
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>License ARN.</p>",
                        "smithy.api#required": {}
                    }
                },
                "GranteePrincipalArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {

"smithy.api#documentation": "<p>The grantee principal ARN.</p>",
                        "smithy.api#required": {}
                    }
                },
                "HomeRegion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                        "smithy.api#required": {}
                    }
                },
                "GrantStatus": {
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>",
                        "smithy.api#required": {}
                    }
                },
                "StatusReason": {
                    "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status reason.</p>"
```

```
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Granted operations.</p>",
            "smithy.api#required": {}
          }
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
            "smithy.api#documentation": "<p>The options specified for the grant.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Describes a grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#GrantList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Grant"
      }
    },
    "com.amazonaws.licensemanager#GrantStatus": {
      "type": "enum",
      "members": {
        "PENDING_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_WORKFLOW"
          }
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_ACCEPT"
          }
```

```
        },
        "REJECTED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"

          }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits":
{
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
```

```
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>"
            }

},
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product name.</p>"
            }
          },
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product SKU.</p>"
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#IssuerDetails",
            "traits": {
              "smithy.api#documentation": "<p>Granted license issuer.</p>"
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

          }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>Granted license status.</p>"
            }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
```

```
      "traits": {
        "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
ISO8601-UTC format.</p>"
      }
    },
    "Beneficiary": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
      }
    },
    "Entitlements": {
      "target": "com.amazonaws.licensemanager#EntitlementList",
      "traits": {
        "smithy.api#documentation": "<p>License entitlements.</p>"
      }
    },
    "ConsumptionConfiguration": {
      "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
      "traits": {
        "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
      }
    },
    "LicenseMetadata": {
      "target": "com.amazonaws.licensemanager#MetadataList",
      "traits": {
        "smithy.api#documentation": "<p>Granted license metadata.</p>"
      }
    },
    "CreateTime": {
      "target": "com.amazonaws.licensemanager#ISO8601DateTime",
      "traits": {
        "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

  }
    },
    "Version": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Version of the granted license.</p>"
      }
    },
    "ReceivedMetadata": {
      "target": "com.amazonaws.licensemanager#ReceivedMetadata",
      "traits": {
        "smithy.api#documentation": "<p>Granted license received metadata.</p>"
      }
    }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
        }
      },
      "com.amazonaws.licensemanager#GrantedLicenseList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#GrantedLicense"
        }
      },
      "com.amazonaws.licensemanager#ISO8601DateTime":
{
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-
1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
        }
      },
      "com.amazonaws.licensemanager#Integer": {
        "type": "integer"
      },
      "com.amazonaws.licensemanager#InvalidParameterValueException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidParameterValueProvided",
            "httpResponseCode": 400
          },
          "smithy.api#documentation":
"<p>One or more parameter values are not valid.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#InvalidResourceStateException": {
        "type": "structure",
        "members": {
          "Message": {
```

```
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InvalidResourceState",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },

    "com.amazonaws.licensemanager#InventoryFilter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Condition": {
          "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
          "traits": {
            "smithy.api#documentation": "<p>Condition of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }

    },
      "traits": {
        "smithy.api#documentation": "<p>An inventory filter.</p>"
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterCondition": {
      "type": "enum",
      "members": {
```

```
      "EQUALS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "EQUALS"
        }
      },
      "NOT_EQUALS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "NOT_EQUALS"
        }
      },
      "BEGINS_WITH": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "BEGINS_WITH"
        }
      },
      "CONTAINS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "CONTAINS"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#InventoryFilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#InventoryFilter"
    }
  },
  "com.amazonaws.licensemanager#Issuer": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Issuer name.</p>",
          "smithy.api#required": {}
        }
      },
      "SignKey": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
```

of sign and verify, \n   and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
            }
          }
        },
        "com.amazonaws.licensemanager#IssuerDetails": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Issuer name.</p>"
              }
            },
            "SignKey": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS key must have a key usage of sign and verify, \n   and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
              }
            },
            "KeyFingerprint": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
          }
        },
        "com.amazonaws.licensemanager#License": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
              }
            },
            "LicenseName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {

```
          "smithy.api#documentation": "<p>License name.</p>"
        }
      },
      "ProductName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product name.</p>"
        }
      },
      "ProductSKU": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product SKU.</p>"
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#IssuerDetails",
        "traits": {
          "smithy.api#documentation": "<p>License issuer.</p>"
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {

"smithy.api#documentation": "<p>Home Region of the license.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
        "traits": {
          "smithy.api#documentation": "<p>License status.</p>"
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License beneficiary.</p>"
        }
      },
```

```
            "Entitlements": {

    "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
                "smithy.api#documentation": "<p>License entitlements.</p>"
            }
        },
        "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
            }
        },
        "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
                "smithy.api#documentation": "<p>License metadata.</p>"
            }
        },
        "CreateTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>License
creation time.</p>"
            }
        },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>License version.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
    }
},
"com.amazonaws.licensemanager#LicenseConfiguration": {
    "type": "structure",
    "members": {
        "LicenseConfigurationId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
            }
        },
        "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
```

```
    "traits": {
      "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
    }
},
"Name": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>Name of the license configuration.</p>"
  }
},
"Description": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>Description of the license configuration.</p>"
  }
},
"LicenseCountingType": {
  "target": "com.amazonaws.licensemanager#LicenseCountingType",
  "traits": {
    "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
  }
},
"LicenseRules": {
  "target": "com.amazonaws.licensemanager#StringList",
  "traits": {
    "smithy.api#documentation": "<p>License rules.</p>"
  }
},
"LicenseCount": {
  "target": "com.amazonaws.licensemanager#BoxLong",
  "traits": {
    "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
  }
},
"LicenseCountHardLimit": {
  "target": "com.amazonaws.licensemanager#BoxBoolean",
  "traits": {
    "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
  }
},
"DisassociateWhenNotFound": {
  "target": "com.amazonaws.licensemanager#BoxBoolean",
  "traits": {
    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
  }
},
"ConsumedLicenses": {
```

```
      "target": "com.amazonaws.licensemanager#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Status of the license configuration.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

      }
    },
    "ConsumedLicenseSummaryList": {
      "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
      }
    },
    "ManagedResourceSummaryList": {
      "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
      }
    },
    "ProductInformationList": {
      "target": "com.amazonaws.licensemanager#ProductInformationList",
      "traits": {
        "smithy.api#documentation": "<p>Product information.</p>"
      }
    },
    "AutomatedDiscoveryInformation": {

      "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
      "traits": {
        "smithy.api#documentation": "<p>Automated discovery information.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n        consumed and enforced by License Manager. Components include specifications for
the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
```

Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n   must be associated with a host), and the number of licenses purchased and used.</p>"
            }
        },
        "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
            "type": "structure",
            "members": {
                "ResourceArn":
{
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
                    }
                },
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#ResourceType",
                    "traits": {
                        "smithy.api#documentation": "<p>Type of server resource.</p>"
                    }
                },
                "ResourceOwnerId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource consuming licenses.</p>"
                    }
                },
                "AssociationTime": {
                    "target": "com.amazonaws.licensemanager#DateTime",
                    "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
                    }
                },
                "AmiAssociationScope": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-account</code>.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
            "type": "list",

```
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
        "type": "enum",
        "members":
{
          "AVAILABLE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "AVAILABLE"
            }
          },
          "DISABLED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DISABLED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits":
{
              "smithy.api#documentation": "<p>Type of resource.</p>"
            }
          },
          "ResourceStatus": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Status of the resource.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
```

```
            }
          },
          "AssociationTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
            }
          },

    "ConsumedLicenses": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurations": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfiguration"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
        "members": {
          "UsageOperation": {
            "target": "com.amazonaws.licensemanager#UsageOperation",
            "traits": {
              "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n       </a>\n        </p>"
            }
          }
        },
```

```
        "traits": {
          "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {

    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the conversion task.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
```

```
        "StartTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation":
 "<p>The time the conversion task was started at.</p>"
          }
        },
        "LicenseConversionTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
          }
        },
        "EndTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskId": {
      "type": "string",
      "traits": {


      "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
      "type": "enum",
      "members": {
        "IN_PROGRESS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "IN_PROGRESS"
          }
        },
        "SUCCEEDED": {
          "target": "smithy.api#Unit",
          "traits": {
```

```
              "smithy.api#enumValue": "SUCCEEDED"
            }
          },
          "FAILED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTasks":
{
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTask"
        }
      },
      "com.amazonaws.licensemanager#LicenseCountingType": {
        "type": "enum",
        "members": {
          "VCPU": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "vCPU"
            }
          },
          "INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Instance"
            }
          },
          "CORE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Core"
            }
          },
          "SOCKET": {
            "target": "smithy.api#Unit",
             "traits": {
              "smithy.api#enumValue": "Socket"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseDeletionStatus": {
```

```
        "type": "enum",
        "members": {
            "PENDING_DELETE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "PENDING_DELETE"
                }
            },
            "DELETED": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DELETED"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#LicenseList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#License"
        }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailure": {

        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
                }
            },
            "ResourceType": {
                "target": "com.amazonaws.licensemanager#ResourceType",
                "traits": {
                    "smithy.api#documentation": "<p>Resource type.</p>"
                }
            },
            "ErrorMessage": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Error message.</p>"
                }
            },
            "FailureTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
```

```
            "smithy.api#documentation": "<p>Failure time.</p>"
          }
        },
        "OperationName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the operation.</p>"
          }
        },
        "ResourceOwnerId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
          }
        },
        "OperationRequestedBy": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
          }
        },
        "MetadataList": {

          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Reserved.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailureList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
      }
    },
    "com.amazonaws.licensemanager#LicenseSpecification": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "AmiAssociationScope": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseSpecifications": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseSpecification"
      }
    },
    "com.amazonaws.licensemanager#LicenseStatus": {
      "type": "enum",
      "members": {
        "AVAILABLE": {
          "target": "smithy.api#Unit",

          "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "PENDING_AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_AVAILABLE"
          }
        },
        "DEACTIVATED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DEACTIVATED"
          }
        },
        "SUSPENDED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUSPENDED"
```

```
          }
        },
        "EXPIRED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EXPIRED"
          }
        },

    "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseUsage": {
      "type": "structure",
      "members": {
        "EntitlementUsages": {
          "target": "com.amazonaws.licensemanager#EntitlementUsageList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlement usages.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

      }
    },
    "com.amazonaws.licensemanager#LicenseUsageException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "LicenseUsageFailure",
```

```
          "httpResponseCode": 412
        },
        "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 412
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
      },
      "output": {

    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
```

```
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
        "type": "structure",
        "members": {
          "LicenseConfigurationAssociations": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
            "traits": {
              "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
            }
          },
          "NextToken": {

"target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListDistributedGrants": {
        "type": "operation",
```

```
            "input": {
               "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
            },
            "output": {
               "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
            },
            "errors": [
               {
                  "target": "com.amazonaws.licensemanager#AccessDeniedException"
               },
               {
                  "target": "com.amazonaws.licensemanager#AuthorizationException"
               },
               {
                  "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
               {
                  "target": "com.amazonaws.licensemanager#RateLimitExceededException"
               },
               {
                  "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
               },
               {
                  "target": "com.amazonaws.licensemanager#ServerInternalException"
               },
               {
                  "target": "com.amazonaws.licensemanager#ValidationException"
               }
            ],
            "traits": {
               "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
            }
         },
         "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
            "type": "structure",
            "members": {
               "GrantArns": {
                  "target": "com.amazonaws.licensemanager#ArnList",
                  "traits": {

               "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
                  }
               },
               "Filters": {
                  "target": "com.amazonaws.licensemanager#FilterList",
                  "traits": {
                     "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>LicenseArn</code>\n            </p>\n
```

          \n   \n   <p>\n   <code>GrantStatus</code>\n   </p>\n   \n
     \n   <p>\n   <code>GranteePrincipalARN</code>\n   </p>\n   \n
\n   <p>\n   <code>ProductSKU</code>\n   </p>\n   \n   \n
<p>\n   <code>LicenseIssuerName</code>\n
         </p>\n   \n   "
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
      "type": "structure",
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",

    "traits": {
            "smithy.api#documentation": "<p>Distributed grant details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
      "type": "operation",
      "input": {

```
            "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
        },
        "errors": [
            {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
```

```
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
        "type": "structure",
        "members": {
          "LicenseOperationFailureList": {
            "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
            "traits": {
              "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
        },

    "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license configurations
 for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArns": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#Filters",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n      are supported:</p>\n      <ul>\n        <li>\n          <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n          Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n          Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n
<p>\n            <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n          Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
```

\n   \n   <p>\n   <code>usagelimitExceeded</code> - A Boolean value that indicates whether the available licenses have been exceeded. \n   Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   "
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
        "type": "structure",
        "members": {
            "LicenseConfigurations": {
                "target": "com.amazonaws.licensemanager#LicenseConfigurations",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license configurations.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",

                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"

        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
      "type": "structure",
      "members": {
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#Filters",
          "traits": {
            "smithy.api#documentation": "<p>\n   Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n   </p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
      "type": "structure",
      "members": {
        "LicenseConversionTasks": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configuration tasks for your

```
account.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target":
"com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
```

```
          ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n    <ul>\n        <li>\n            <p>\n              <code>LicenseConfigurationArn</code>\n
</p>\n        </li>\n    </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits":
{
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
        "type": "structure",
        "members": {
          "ReportGenerators": {
            "target": "com.amazonaws.licensemanager#ReportGeneratorList",
            "traits": {
              "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }

        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
            "smithy.api#required": {}
          }
```

```
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {

        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
        "type": "structure",
        "members": {
            "LicenseSpecifications": {
                "target": "com.amazonaws.licensemanager#LicenseSpecifications",
                "traits": {
                    "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseVersions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
        },
        "errors": [
```

```
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
```

```
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#LicenseList",
            "traits": {
              "smithy.api#documentation": "<p>License details.</p>"
            }
          },

"NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenses": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicensesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicensesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
```

```
      "traits": {
        "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },

    "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>Beneficiary</code>\n            </p>\n
          </li>\n          <li>\n            <p>\n              <code>ProductSKU</code>\n            </p>\n          </li>\n
      <li>\n            <p>\n              <code>Fingerprint</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>Status</code>\n            </p>\n          </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token
for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicensesResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#LicenseList",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>License details.</p>"
                  }
                },
                "NextToken": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {

"smithy.api#documentation": "<p>Token for the next set of results.</p>"
                  }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedGrants": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
```

specifying the\n   recipient as this Amazon Web Services account, your organization, or an organizational unit\n
(OU) to which this member account belongs.</p>"
              }
          },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target":
"com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
                        "smithy.api#required": {}

```
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
      <ul>\n         <li>\n            <p>\n               <code>ParentArn</code>\n            </p>\n         </li>\n
<li>\n            <p>\n               <code>GranteePrincipalArn</code>\n            </p>\n         </li>\n      </ul>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
        "type": "structure",
        "members": {
            "Grants": {
                "target": "com.amazonaws.licensemanager#GrantList",
                "traits": {
                    "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
```

```
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
            "type": "structure",
            "members": {
                "GrantArns": {
                    "target": "com.amazonaws.licensemanager#ArnList",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n               <code>ProductSKU</code>\n            </p>\n
          </li>\n          <li>\n            <p>\n               <code>LicenseIssuerName</code>\n            </p>\n
</li>\n          <li>\n            <p>\n               <code>LicenseArn</code>\n            </p>\n          </li>\n
<li>\n            <p>\n               <code>GrantStatus</code>\n            </p>\n          </li>\n          <li>\n
<p>\n
      <code>GranterAccountId</code>\n            </p>\n          </li>\n        </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
            "type": "structure",
            "members": {
                "Grants": {
                    "target": "com.amazonaws.licensemanager#GrantList",
                    "traits": {
                        "smithy.api#documentation": "<p>Received grant details.</p>"
                    }
                },
```

```
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicenses": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists received licenses.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
```

```
{
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {

    "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
    }
},
"com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>Beneficiary</code>\n                </p>\n
    </li>\n            <li>\n                <p>\n                    <code>ProductSKU</code>\n
        </p>\n            </li>\n        </ul>"
            }
        },
```

```
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#GrantedLicenseList",
          "traits": {
            "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },
```

```
      "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
               "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n      <ul>\n         <li>\n            <p>\n               <code>ProductSKU</code>\n            </p>\n
      </li>\n         <li>\n            <p>\n               <code>Status</code>\n            </p>\n         </li>\n
<li>\n         <p>\n            <code>Fingerprint</code>\n            </p>\n         </li>\n         <li>\n
<p>\n               <code>IssuerName</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
<code>Beneficiary</code>\n            </p>\n         </li>\n      </ul>"
            }
         },
         "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
             "traits": {
               "smithy.api#documentation": "<p>Token for the next set of results.</p>"
             }
         },
         "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
               "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
         }
      },
      "traits": {
         "smithy.api#input": {}
      }
   },
   "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
      "type": "structure",
      "members": {
         "Licenses": {
            "target": "com.amazonaws.licensemanager#GrantedLicenseList",
            "traits": {
               "smithy.api#documentation": "<p>Received license details.</p>"
            }
         },

         "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
               "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
         }
      },
      "traits": {
         "smithy.api#output": {}
      }
```

        },
        "com.amazonaws.licensemanager#ListResourceInventory": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#FailedDependencyException"
                },
                {
                    "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
            "type": "structure",
            "members": {
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {

```
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanager#InventoryFilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n       are supported:</p>\n       <ul>\n          <li>\n             <p>\n                <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n         Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
          </li>\n          <li>\n             <p>\n                <code>application_name</code> - The name of the
application. \n          Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
   </li>\n          <li>\n             <p>\n                <code>license_included</code> - The type of license included. \n
       Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n          Possible values
are <code>sql-server-enterprise</code> | \n          <code>sql-server-standard</code> | \n          <code>sql-
server-web</code> | \n          <code>windows-server-datacenter</code>.</p>\n          </li>\n          <li>\n
   <p>\n                <code>platform</code> - The platform of the resource.\n          Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n          </li>\n          <li>\n             <p>\n
       <code>resource_id</code> - The ID of the resource. \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n          <li>\n             <p>\n
   <code>tag:<key></code> - The key/value combination of a tag assigned \n          to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n          <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n          </li>\n       </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
        "type": "structure",
        "members": {
            "ResourceInventoryList": {
                "target": "com.amazonaws.licensemanager#ResourceInventoryList",
                "traits": {
                    "smithy.api#documentation": "<p>Information
about the resources.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
```

```
          "smithy.api#output": { }
        }
      },
      "com.amazonaws.licensemanager#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

    "smithy.api#required": { }
          }
        }
      },
      "traits": {
        "smithy.api#input": { }
      }
```

```
    },
    "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the tags.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListTokens": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListTokensRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListTokensResponse"
      },
      "errors":
[
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists your tokens.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTokensRequest": {
      "type": "structure",
      "members": {
```

```
        "TokenIds": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {


            "smithy.api#documentation": "<p>Token IDs.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n        <li>\n            <p>\n                <code>LicenseArns</code>\n            </p>\n        </li>\n
</ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListTokensResponse": {
      "type": "structure",
      "members": {
        "Tokens": {
          "target": "com.amazonaws.licensemanager#TokenList",
          "traits": {
            "smithy.api#documentation": "<p>Received token details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
```

```
        },
        "traits": {
          "smithy.api#output": { }
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
  {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n        consumption details by resource at a selected point in time. Use this action to audit the\n        current
license consumption for any license inventory and configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#Filters",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n      are supported:</p>\n      <ul>\n        <li>\n          <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
     Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n
 <li>\n          <p>\n              <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n          <p>\n
  <code>resourceAccount</code> - The ID of the account that owns the resource. \n          Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n      </ul>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {

"LicenseConfigurationUsageList": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#Location": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#Long": {
        "type": "long"
      },
      "com.amazonaws.licensemanager#ManagedResourceSummary": {
        "type": "structure",
        "members": {
          "ResourceType":
{
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
            }
          },
          "AssociationCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
        }
      },
      "com.amazonaws.licensemanager#MaxSize100": {

        "type": "integer",
        "traits": {
          "smithy.api#range": {
            "min": 1,
            "max": 100
```

```
            }
          }
        },
        "com.amazonaws.licensemanager#MaxSize3StringList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 3
            }
          }
        },
        "com.amazonaws.licensemanager#Message": {
          "type": "string"
        },
        "com.amazonaws.licensemanager#Metadata": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The key name.</p>"
              }
            },

        "Value": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
          }
        },
        "com.amazonaws.licensemanager#MetadataList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Metadata"
          }
        },
        "com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
          "type": "structure",
          "members": {
```

```
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#Options": {
    "type": "structure",
    "members": {
      "ActivationOverrideBehavior": {
        "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
        "traits": {
          "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n        Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n        <ul>\n            <li>\n                <p>As a license administrator, you can
optionally specify an\n                <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
</li>\n            <li>\n                <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n                account in your
organization.</p>\n            </li>\n            <li>\n                <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n                <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n             are activating the grant.</p>\n            </li>\n        </ul>\n        <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n            <dd>\n                <p>Use this value to activate a grant
without replacing any member accounts\n                active grants for the same product.</p>\n            </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n            <dd>\n                <p>Use this value to activate a
grant and disable other active grants in any\n              member accounts
 for the same product. This action will also replace their\n                 previously activated grants with this activated
grant.</p>\n            </dd>\n        </dl>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n        override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager</a> in the <i>License Manager User Guide</i>.</p>"
    }
  },
  "com.amazonaws.licensemanager#OrganizationConfiguration": {
    "type": "structure",
    "members": {
      "EnableIntegration": {
        "target": "com.amazonaws.licensemanager#Boolean",
```

```
                    "traits": {
                        "smithy.api#default": false,
                        "smithy.api#documentation":
"<p>Enables Organizations integration.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
            }
        },
        "com.amazonaws.licensemanager#PrincipalArnList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#Arn"
            },
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 1
                }
            }
        },
        "com.amazonaws.licensemanager#ProductInformation": {
            "type": "structure",
            "members": {
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Resource type. The
possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ProductInformationFilterList": {
                    "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n        <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n        <p>The following filters and are supported when
the resource type \n        is <code>SSM_MANAGED</code>:</p>\n
        <ul>\n        <li>\n          <p>\n            <code>Application Name</code> - The name of the
application.\n          Logical operator is <code>EQUALS</code>.</p>\n          </li>\n        <li>\n
<p>\n          <code>Application Publisher</code> - The publisher of the application.\n          Logical
```

operator is <code>EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>Application Version</code> - The version of the application.\n   Logical operator is <code>EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>Platform Name</code> - The name of the platform.\n   Logical operator is <code>EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>Platform Type</code> - The platform type.\n   Logical operator is <code>EQUALS</code>.</p>\n

        \n   \n   <p>\n   <code>Tag:key</code> - The key of a tag attached to an Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is <code>NOT_EQUALS</code>. The key for your tag must be appended to <code>Tag:</code> following the example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you are not using values for the key.\n   </p>\n   \n   \n   <p>\n   <code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from automated discovery.\n   Logical operator is <code>NOT_EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>License Included</code> - The type of license included.\n   Logical operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n   Possible values are:\n   <code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-server-web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n   <p>The following filters and logical operators are supported when the resource type\n   is <code>RDS</code>:</p>\n   \n   \n   <p>\n   <code>Engine Edition</code> - The edition of the database engine.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-se2</code>.</p>\n   \n   \n   <p>\n   <code>License Pack</code> - The license pack.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>data guard</code> | \n   <code>diagnostic pack sqlt</code> | \n   <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n   <code>olap</code>.</p>\n   \n   ",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#ProductInformationFilter": {
          "type": "structure",
          "members": {
            "ProductInformationFilterName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Filter name.</p>",
                "smithy.api#required": {}
              }
            },
            "ProductInformationFilterValue": {
              "target": "com.amazonaws.licensemanager#StringList",

```
      "traits": {
            "smithy.api#documentation": "<p>Filter value.</p>"
        }
      },
      "ProductInformationFilterComparator": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Logical operator.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes product information filters.</p>"
    }
  },
  "com.amazonaws.licensemanager#ProductInformationFilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ProductInformationFilter"
    }
  },
  "com.amazonaws.licensemanager#ProductInformationList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ProductInformation"
    }
  },
  "com.amazonaws.licensemanager#ProvisionalConfiguration": {
    "type": "structure",
    "members": {
      "MaxTimeToLiveInMinutes": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#RateLimitExceededException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
```

```
                    }
                },
                "traits": {
                    "aws.protocols#awsQueryError": {
                        "code": "RateLimitExceeded",
                        "httpResponseCode": 429
                    },
                    "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
                    "smithy.api#error": "client",
                    "smithy.api#httpError": 429
                }
            },
            "com.amazonaws.licensemanager#ReceivedMetadata": {
                "type": "structure",
                "members": {
                    "ReceivedStatus": {
                        "target": "com.amazonaws.licensemanager#ReceivedStatus",
                        "traits": {
                            "smithy.api#documentation": "<p>Received status.</p>"
                        }
                    },
                    "ReceivedStatusReason": {
                        "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                        "traits": {
                            "smithy.api#documentation": "<p>Received
status reason.</p>"
                        }
                    },
                    "AllowedOperations": {
                        "target": "com.amazonaws.licensemanager#AllowedOperationList",
                        "traits": {
                            "smithy.api#documentation": "<p>Allowed operations.</p>"
                        }
                    }
                },
                "traits": {
                    "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
                }
            },
            "com.amazonaws.licensemanager#ReceivedStatus": {
                "type": "enum",
                "members": {
                    "PENDING_WORKFLOW": {
                        "target": "smithy.api#Unit",
                        "traits": {
                            "smithy.api#enumValue": "PENDING_WORKFLOW"
                        }
```

```
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"PENDING_ACCEPT"
          }
        },
        "REJECTED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"
          }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
           "traits": {
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#RedirectException": {
      "type": "structure",
```

```
      "members": {
        "Location": {
          "target": "com.amazonaws.licensemanager#Location",
          "traits": {
            "smithy.api#httpHeader": "Location"
          }
        },
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 308
      }
    },
    "com.amazonaws.licensemanager#RejectGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#RejectGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#RejectGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#RejectGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#RejectGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#RenewType":
```

```
{
      "type": "enum",
      "members": {
         "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "None"
            }
         },
         "WEEKLY": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Weekly"
            }
         },
         "MONTHLY": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Monthly"
            }
         }
      }
   },
   "com.amazonaws.licensemanager#ReportContext": {
      "type": "structure",
      "members": {
         "licenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {

      "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
               "smithy.api#required": {}
            }
         }
      },
      "traits": {
         "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
      }
   },
   "com.amazonaws.licensemanager#ReportFrequency": {
      "type": "structure",
      "members": {
         "value": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
               "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
```

```
              }
            },
            "period": {
              "target": "com.amazonaws.licensemanager#ReportFrequencyType",

              "traits": {
                "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
          }
        },
        "com.amazonaws.licensemanager#ReportFrequencyType": {
          "type": "enum",
          "members": {
            "DAY": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "DAY"
              }
            },
            "WEEK": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "WEEK"
              }
            },
            "MONTH": {
              "target": "smithy.api#Unit",
              "traits":
{
                "smithy.api#enumValue": "MONTH"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#ReportGenerator": {
          "type": "structure",
          "members": {
            "ReportGeneratorName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>"
              }
            },
```

```
        "ReportType": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
          }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
          }
        },
        "ReportFrequency": {
          "target": "com.amazonaws.licensemanager#ReportFrequency",
          "traits": {
            "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
          }
        },
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
          }
        },
        "LastRunStatus": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
          }
        },
        "LastRunFailureReason":
{
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
          }
        },
        "LastReportGenerationTime": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
          }
        },
        "ReportCreatorAccount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
```

generator.</p>"
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",

     "traits": {
               "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
          },
          "S3Location": {
            "target": "com.amazonaws.licensemanager#S3Location",
            "traits": {
              "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
            }
          },
          "CreateTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Time the report was created.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
            }

    }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
        }
      },
      "com.amazonaws.licensemanager#ReportGeneratorList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#ReportGenerator"
        }
      },
      "com.amazonaws.licensemanager#ReportGeneratorName": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          }

```
        }
      },
      "com.amazonaws.licensemanager#ReportType": {
        "type": "enum",
        "members": {
          "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
            }

          },
          "LICENSE_CONFIGURATION_USAGE_REPORT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "LicenseConfigurationUsageReport"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#ReportTypeList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#ReportType"
        }
      },
      "com.amazonaws.licensemanager#ResourceInventory": {
        "type": "structure",
        "members": {
          "ResourceId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits":
{
              "smithy.api#documentation": "<p>Type of resource.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
```

```
        "Platform": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Platform of the resource.</p>"
          }
        },
        "PlatformVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
          }
        },
        "ResourceOwningAccountId":
{
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ResourceInventoryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ResourceInventory"
      }
    },
    "com.amazonaws.licensemanager#ResourceLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "ResourceLimitExceeded",

          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
```

```
"com.amazonaws.licensemanager#ResourceNotFoundException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
    },
    "traits": {
        "aws.protocols#awsQueryError": {
            "code": "InvalidResource.NotFound",
            "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>The resource cannot be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
    }
},
"com.amazonaws.licensemanager#ResourceType": {
    "type": "enum",
    "members": {
        "EC2_INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "EC2_INSTANCE"
            }
        },
        "EC2_HOST": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "EC2_HOST"
            }
        },
        "EC2_AMI": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "EC2_AMI"
            }
        },
        "RDS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "RDS"
            }
        },
        "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
```

```
                    "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#S3Location": {
        "type": "structure",
        "members": {
            "bucket": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
                }
            },
            "keyPrefix": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
        }
    },
    "com.amazonaws.licensemanager#ServerInternalException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "InternalError",
                "httpResponseCode": 500
            },
            "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanager#SignedToken": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 4096
```

```
            }
          }
        },
        "com.amazonaws.licensemanager#StatusReasonMessage": {
          "type": "string",
          "traits":
{
            "smithy.api#length": {
              "min": 0,
              "max": 400
            },
            "smithy.api#pattern": "^[\\s\\S]+$"
          }
        },
        "com.amazonaws.licensemanager#String": {
          "type": "string"
        },
        "com.amazonaws.licensemanager#StringList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#String"
          }
        },
        "com.amazonaws.licensemanager#Tag": {
          "type": "structure",
          "members": {
            "Key": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Tag key.</p>"
              }
            },
            "Value": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation":
"<p>Tag value.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#TagKeyList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#String"
```

```
        }
      },
      "com.amazonaws.licensemanager#TagList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Tag"
        }
      },
      "com.amazonaws.licensemanager#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#TagResourceResponse"
        },
        "errors": [
          {
            "target":
"com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits":
{
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
```

```
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>One or more tags.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#TokenData": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>"
            }
          },
          "TokenType": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "ExpirationTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
```

format.</p>"
                }
            },
            "TokenProperties": {
                "target": "com.amazonaws.licensemanager#MaxSize3StringList",
                "traits": {
                    "smithy.api#documentation": "<p>Data specified by the caller.</p>"
                }
            },
            "RoleArns": {
                "target": "com.amazonaws.licensemanager#ArnList",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
                }
            }
        },
        "traits": {

            "smithy.api#documentation": "<p>Describes a token.</p>"
        }
    },
    "com.amazonaws.licensemanager#TokenList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#TokenData"
        }
    },
    "com.amazonaws.licensemanager#TokenString": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 4096
            },
            "smithy.api#pattern": "^\\S+$"
        }
    },
    "com.amazonaws.licensemanager#TokenType": {
        "type": "enum",
        "members": {

```
      "REFRESH_TOKEN": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "REFRESH_TOKEN"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
{
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
       },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
```

```
      "traits": {
        "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanager#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
      },
      "output": {

 "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
```

```
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "LicenseConfigurationStatus": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
              "traits": {
                "smithy.api#documentation": "<p>New status of the license configuration.</p>"
              }
            },
            "LicenseRules": {
              "target": "com.amazonaws.licensemanager#StringList",
              "traits": {

  "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
  configuration is licenseAffinityToHost.</p>"
              }
            },
            "LicenseCount": {
              "target": "com.amazonaws.licensemanager#BoxLong",
              "traits": {
```

```
                    "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
                }
            },
            "LicenseCountHardLimit": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
                }
            },
            "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
                }
            },
            "Description": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>New description of the license configuration.</p>"
                }
            },
            "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                    "smithy.api#documentation": "<p>New product information.</p>"
                }
            },
            "DisassociateWhenNotFound": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            }
        },
        "traits":
{
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
```

```
        },
      "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
  {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
```

update.</p>",
                "smithy.api#required": { }
              }
            },
            "ReportGeneratorName": {
              "target": "com.amazonaws.licensemanager#ReportGeneratorName",
              "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>",
                "smithy.api#required": { }
              }
            },
            "Type": {
              "target": "com.amazonaws.licensemanager#ReportTypeList",
              "traits": {
                "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n   \n   \n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n   \n   \n   <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n   \n   ",
                "smithy.api#required": { }
              }
            },
            "ReportContext": {
              "target": "com.amazonaws.licensemanager#ReportContext",
              "traits": {
                "smithy.api#documentation": "<p>The report context.</p>",
                "smithy.api#required": { }
              }
            },
            "ReportFrequency": {
              "target": "com.amazonaws.licensemanager#ReportFrequency",
              "traits":
 {
                "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                "smithy.api#required": { }
              }
            },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientRequestToken",
              "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                "smithy.api#required": { }
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {

```
                    "smithy.api#documentation": "<p>Description of the report generator.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },

            {
                "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
            },
            {
                "target": "com.amazonaws.licensemanager#LicenseUsageException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
```

```
      "traits": {
          "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
  as they send license configurations to the operation that creates the resource.</p>"
      }
    },

    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
              "smithy.api#required": {}
          }
        },
        "AddLicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
          }
        },
        "RemoveLicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
              "smithy.api#documentation":
  "<p>ARNs of the license configurations to remove.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
```

```
          },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
         },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
          }
        },
        "SnsTopicArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
          }
        },
        "OrganizationConfiguration": {
          "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
```

```
                    }
                },
                "EnableCrossAccountsDiscovery": {
                    "target": "com.amazonaws.licensemanager#BoxBoolean",
                    "traits": {
                        "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#UsageOperation": {
            "type": "string",
            "traits": {
                "smithy.api#length": {
                    "min": 0,
                    "max": 50
                }
            }
        },
        "com.amazonaws.licensemanager#ValidationException": {
            "type": "structure",
            "members": {
                "Message": {
                    "target": "com.amazonaws.licensemanager#Message"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

                "smithy.api#error": "client",
                "smithy.api#httpError": 400
            }
        }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```go
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version
 // of the license.
 //
 // This member is required.
 SourceVersion *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {
```

```go
    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack,
options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
        return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
```

```
    return err
  }
  if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
  }
  if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
  }
  if err = addClientUserAgent(stack, options); err != nil {
    return err
  }
  if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
  }
  if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
  }
  if
  err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
  }
  if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
    return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
    return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
  }
  return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
  return &awsmiddleware.RegisterServiceMetadata{
```

```
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
  }
  }
```

1.27 aws-aws-sdk-go-v2 1.26.1

1.27.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
 params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
 return nil,
 err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {
```

```go
    // The type of license to associate with the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err !=
    nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
    return err
  }
  if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
   return err
  }
  if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
   return err
  }
  if err = addResolveEndpointMiddleware(stack, options); err != nil {
   return err
  }
  if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
   return err
  }
  if err = addRetryMiddlewares(stack, options); err != nil {
   return err
  }
  if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
   return err
  }
  if err = awsmiddleware.AddRecordResponseTiming(stack);
  err != nil {
   return err
  }
  if err = addClientUserAgent(stack, options); err != nil {
   return err
  }
  if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
   return err
  }
  if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
   return err
  }
  if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
   return err
  }
  if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
   return err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
   return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
   return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
   return err
```

```
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack,
    options); err != nil {
     return err
    }
    return nil
    }


    func newServiceMetadataMiddleware_opAssociateLicense(region string)
    *awsmiddleware.RegisterServiceMetadata {
     return &awsmiddleware.RegisterServiceMetadata{
      Region:       region,
      ServiceID:     ServiceID,
      OperationName: "AssociateLicense",
     }
    }
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

```
name: License Scan

on: [pull_request]

jobs:
 licensescan:
  name: License Scan
  runs-on: ubuntu-latest
  strategy:
   matrix:
    python-version: [3.9]

  steps:
   - name: Checkout target
    uses: actions/checkout@v2
    with:
     path: sdkbase
     ref: ${{ github.base_ref }}
   - name: Checkout this ref
    uses: actions/checkout@v2
    with:
     path: new-ref
     fetch-depth: 0
```

```yaml
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)


// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }
```

```go
	result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*CheckInLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type CheckInLicenseInput struct {

	// License consumption token.
	//
	// This member is required.
	LicenseConsumptionToken *string

	// License beneficiary.
	Beneficiary
	*string

	noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return
err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
```

```
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckInLicense",
	}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
            "type": "structure",
            "members": {
                "Name": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
                    }
                },
                "Values": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                    "traits": {
```

```
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n     Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
```

```
            "smithy.api#idempotent": {}
          }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
          "type": "structure",
          "members": {}
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n      Organizations integration has been enabled, and
 which Regions data will be aggregated from.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays
the aggregated data for Linux\n      subscriptions.</p>"
              }
            }
          }
      },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }

    },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        },
        "Region": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
            }
```

```
            },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
              "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
          },
          "LastUpdatedTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
```

```
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
    ],

    "traits": {
        "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
            "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                "x-api-key",
                "authorization",
                "x-amz-date",

"x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
```

```
                ]
              },
            "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager Linux Subscriptions",
            "smithy.rules#endpointRuleSet": {
                "version": "1.0",
                "parameters": {
                  "Region": {
                      "builtIn": "AWS::Region",
                      "required": false,
                      "documentation": "The
 AWS region used to dispatch the request.",
                      "type": "String"
                  },
                  "UseDualStack": {
                      "builtIn": "AWS::UseDualStack",
                      "required": true,
                      "default": false,
                      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                      "type": "Boolean"
                  },
                  "UseFIPS": {
                      "builtIn": "AWS::UseFIPS",
                      "required": true,
                      "default": false,
                      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                      "type": "Boolean"
                  },
                  "Endpoint": {
                      "builtIn": "SDK::Endpoint",
                      "required": false,
                      "documentation": "Override the endpoint used to send this request",
                      "type": "String"
                  }
                },
                "rules": [
                  {
                      "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                              {
                                  "ref": "Endpoint"
                              }
```

```
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {

                "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],
                    "type": "tree",
                    "rules": [
                        {

                    "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
                                        },
                                        true
                                    ]
                                }
                            ],
                            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                            "type": "error"
                        },
                        {
                            "conditions": [],

                "endpoint": {
                                "url": {
                                    "ref": "Endpoint"
                                },
```

```
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }
              ]
            }
          ]
        },
        {
          "conditions": [],
          "type": "tree",
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",

    "argv": [
                    {
                      "ref": "Region"
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "aws.partition",
                      "argv": [
                        {
                          "ref": "Region"
                        }
                      ],

  "assign": "PartitionResult"
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
```

```
                              "ref": "UseFIPS"
                        },
                        true
                      ]
                  },

              {
                      "fn": "booleanEquals",
                    "argv": [
                      {
                          "ref": "UseDualStack"
                      },
                        true
                    ]
                }
              ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {

                "fn": "booleanEquals",
                      "argv": [
                          true,
                        {
                            "fn": "getAttr",
                            "argv": [
                              {
                                  "ref": "PartitionResult"
                              },
                                "supportsFIPS"
                            ]

    }

                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                              "fn": "getAttr",
                              "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
```

```
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [],
                            "type": "tree",
                            "rules": [
                                {

                                    "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                }
                            ]
                        }
                    ]

                },
                {
                    "conditions": [],
                    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                    "type": "error"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {

                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
```

```
            ],
            "type": "tree",
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,

    {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      }
                    ]
                  }
                ],
                "type": "tree",

                "rules": [
                  {
                    "conditions": [],
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                          "properties": {},
                          "headers": {}

                        },
                        "type": "endpoint"
                      }
                    ]
                  }
                ]
              },
              {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"
```

```
                    }
                ]

        },
                            {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {

                         "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                            ]
                                        }
                                    ]
                                }
                            ],
                            "type": "tree",
                            "rules": [
                                {
                                    "conditions": [],
                                    "type": "tree",
                                    "rules": [
                                        {
                                            "conditions": [],

                                    "endpoint": {
                                            "url": "https://license-manager-linux-
```

subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                          "properties": {},
                                          "headers": {}
                                    },
                                    "type": "endpoint"
                              }
                        ]
                  }
              ]
          },

{

                              "conditions": [],
                              "error": "DualStack is enabled but this partition does not support DualStack",
                              "type": "error"
                        }
                  ]
            },
            {
                  "conditions": [],
                  "type": "tree",
                  "rules": [
                        {
                              "conditions": [],
                              "endpoint": {
                                    "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                              },
                              "type": "endpoint"
                        }
                  ]
            }
        ]
      }
    ]
  },
  {
      "conditions": [],
      "error": "Invalid Configuration: Missing Region",
      "type": "error"
  }
]
}
]
},
"smithy.rules#endpointTests": {

```
"testCases": [
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    },
```

```
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {

        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
```

```
        "expect": {
          "endpoint":
{
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
           "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint":
{
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
```

```
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
      },
      "params": {
        "Region": "us-iso-east-1",
```

```
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {

  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {

         "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
          }
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
```

```
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",

            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",

          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
```

```
                    }
                  },
                  {
                    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                    "expect": {
                      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false,

              "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "Missing region",
                    "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }

          }
            ],
              "version": "1.0"
          }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled LinuxSubscriptionsDiscovery"
          },
```

```
          {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled LinuxSubscriptionsDiscovery"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
      "type": "structure",
      "members": {
        "SourceRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "OrganizationIntegration": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
          "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
      "type":
"operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with
commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {

    "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by
 the name of\n      <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n      <p>The valid names for this filter
are:</p>\n      <ul>\n         <li>\n            <p>\n               <code>AmiID</code>\n            </p>\n
</li>\n         <li>\n            <p>\n               <code>InstanceID</code>\n            </p>\n         </li>\n
<li>\n            <p>\n               <code>AccountID</code>\n            </p>\n         </li>\n         <li>\n
<p>\n               <code>Status</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
<code>Region</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n         </li>\n         <li>\n
    <p>\n               <code>ProductCode</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
    <code>InstanceType</code>\n            </p>\n         </li>\n      </ul>\n      <p>The valid Operators for this
filter are:</p>\n      <ul>\n         <li>\n            <p>\n               <code>contains</code>\n            </p>\n
</li>\n         <li>\n            <p>\n               <code>equals</code>\n            </p>\n         </li>\n         <li>\n
    <p>\n               <code>Notequal</code>\n            </p>\n         </li>\n      </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
```

```
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 16384
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
        "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
                "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
        {
```

```
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n     organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
            }

        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
              "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n     partially match, or don't
match a certain subscription's name.</p>\n     <p>The valid names for this filter are:</p>\n     <ul>\n
<li>\n           <p>\n              <code>Subscription</code>\n           </p>\n        </li>\n     </ul>\n
<p>The valid Operators for this filter are:</p>\n

     <ul>\n        <li>\n           <p>\n              <code>contains</code>\n           </p>\n        </li>\n
<li>\n           <p>\n              <code>equals</code>\n           </p>\n        </li>\n        <li>\n           <p>\n
    <code>Notequal</code>\n           </p>\n        </li>\n     </ul>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
```

"smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                        "smithy.api#length": {

                            "min": 1,
                            "max": 16384
                        }
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
            "type": "structure",
            "members": {
                "Subscriptions": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                    "traits": {
                        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "EQUAL",
                        "value": "Equal",
                        "documentation": "Equal operator"
                    },
                    {

```json
                    "name": "NOT_EQUAL",
                    "value": "NotEqual",
                    "documentation": "Not equal operator"
                },
                {
                    "name": "CONTAINS",
                    "value": "Contains",
                    "documentation": "Contains operator"
                }
            ],
            "smithy.api#length": {
                "min": 1,
                "max": 20
            }
        }
    },

    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [

{
                "name": "InProgress",
                "value": "InProgress",
```

```json
              "documentation": "InProgress status"
            },
            {
              "name": "Completed",
              "value": "Completed",
              "documentation": "Completed status"
            },
            {
              "name": "Successful",
              "value": "Successful",
              "documentation": "Successful status"
            },
            {
              "name": "Failed",
              "value": "Failed",
              "documentation": "Failed status"
            }
          ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String",
        "traits":
{
          "smithy.api#length": {
            "min": 1,
            "max": 100
          }
        }
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"

        }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },

        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required": {}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
```

```
from.</p>",

                "smithy.api#required": {}
                        }
                },
                "AllowUpdate": {
                    "target": "smithy.api#Boolean",
                    "traits": {
                        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
            "type": "structure",
            "members": {
                "LinuxSubscriptionsDiscovery": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                    "traits": {
                        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                    }
                },

                "LinuxSubscriptionsDiscoverySettings": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                    "traits": {
                        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                    "traits": {
                        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                    }
                },
                "StatusMessage": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                    "traits": {
                        "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
                    }
                },
                "HomeRegions": {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}
}
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
 params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
 return nil, err
 }
```

```go
out := result.(*ListLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //   - Beneficiary
 //   - ProductSKU
 //   - Fingerprint
 //
   - Status
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
```

```go
	return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
	if err != nil {
	return err
	}
	if
	err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
	return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
	return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
	return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
	return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
	err != nil {
	return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
```

```go
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:
			ServiceID,
		OperationName: "ListLicenses",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/grafana/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
```

```go
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 //
 This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
```

```go
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "DisassociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
```

```go
    err != nil {
     return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
     return err
    }
    if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
     return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
 {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
```

```go
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
    c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type
 CheckoutLicenseInput struct {

    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData
```

```go
    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource
    Name (ARN) of the checkout license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata
```

```go
    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil
 {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
```

```go
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack);
	err != nil {
		return err
	}
	if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
	middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```go
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier
 that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see RFC 7518 Digital Signature with
 // RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
```

```go
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata
```

```go
	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
	if err != nil {
		return
	err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
```

```go
	if err = addClientUserAgent(stack, options); err != nil {
		return
err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutBorrowLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```go
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide
 to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
```

```go
	// This member is required.
	ConsumptionConfiguration *types.ConsumptionConfiguration

	// License entitlements.
	//
	// This member is required.
	Entitlements []types.Entitlement

	// Home Region for the license.
	//
	// This member is required.
	HomeRegion *string

	// License issuer.
	//
	// This member is required.
	Issuer *types.Issuer

	// License name.
	//
	// This member is required.
	LicenseName *string

	// Product name.
	//
	// This member is required.
	ProductName *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// Date and time range during which the license is valid, in ISO8601-UTC format.
	//
	// This member is required.
	Validity
	*types.DatetimeRange

	// Information about the license.
	LicenseMetadata []types.Metadata

	noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
```

```go
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add
protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
        return err
```

```
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
        return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
    err != nil {
        return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
        return err
    }
    if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
        return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
        return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
        return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
        return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
        return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
        return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
        return err
    }
    return nil
}
```

```go
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CreateLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
	if params == nil {
		params = &GetLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*GetLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type GetLicenseInput struct {

	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// License version.
```

```go
	Version
	*string

	noSmithyDocumentSerde
}

type GetLicenseOutput struct {

	// License details.
	License *types.License

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return
err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
```

```
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
     return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
     return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
     return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
     return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
     return err
    }
    if err = addOpGetLicenseValidationMiddleware(stack);
    err != nil {
     return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
    err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
```

```go
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
```
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```go
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
	if params == nil {
		params = &ListReceivedLicensesInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
	c.addOperationListReceivedLicensesMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*ListReceivedLicensesOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type ListReceivedLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//
	//   - ProductSKU
	//   - Status
	//   - Fingerprint
	//   - IssuerName
	//   - Beneficiary
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

	// Received license details.
```

```
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
```

```go
    return err
  }
  if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
   return err
  }
  if err = awsmiddleware.AddRecordResponseTiming(stack);
 err != nil {
   return err
  }
  if err = addClientUserAgent(stack, options); err != nil {
   return err
  }
  if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
   return err
  }
  if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
   return err
  }
  if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
   return err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
   return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
   return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
   return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
   return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
   return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
   return err
  }
  return nil
 }

func newServiceMetadataMiddleware_opListReceivedLicenses(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:      region,
  ServiceID:    ServiceID,
```

    OperationName: "ListReceivedLicenses",
  }
}

                     Apache License
                Version 2.0, January 2004
                http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership
   of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or Object
      form, that is based on (or derived from) the Work and for which the
      editorial

revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding
those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions

  for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

 5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {

```
        "smithy.api#documentation":
"<p>Details about an Active Directory identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n        <note>\n          <p>Your estimated bill for charges on the number of users and related
costs will take 48\n       hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
     monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/AssociateUser"
        },
        "smithy.api#idempotent": {}
```

```
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
```

```json
        "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n      resource.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                  "smithy.api#required": {}
               }
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
               "traits": {
                  "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
                  "smithy.api#required": {}
               }
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
         },
         "output": {
```

```
              "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
          }

      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
```

```
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },

        "com.amazonaws.licensemanagerusersubscriptions#Filter": {
          "type": "structure",
          "members": {
            "Attribute": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
              }
            },
            "Operation": {
              "target": "smithy.api#String",
              "traits": {
```

```
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used
to return more specific results from a describe\n      operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n      attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n     settings such as the subnets to provision VPC endpoints.</p>",
            "smithy.api#required":
{}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
```

```
                    "smithy.api#required": {}
                  }
                },
                "Status": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The status
of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                  }
                },
                "Products": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
                  "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
                    "smithy.api#required": {}
                  }
                },
                "LastStatusCheckDate": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                  }
                },
                "StatusMessage": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                  }

        }
            },
          "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
```

```
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits":
  {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
              "smithy.api#required": {}
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          },
          "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
          },
          "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
                }
            }

        },
            "traits": {
                "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
                "smithy.api#error": "server"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
            "type": "service",

        "version": "2018-05-10",
            "operations": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
                },
```

```
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
},
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
}

],
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
```

```
                    "x-amzn-errortype",
                    "x-amzn-requestid",
                    "x-amzn-trace-id"


 ]
          },
          "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager User Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack
 endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "Boolean"
              },
              "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "Boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "String"
              }

          },
            "rules": [
              {
                "conditions": [
                  {
                      "fn": "isSet",
                      "argv": [
```

```
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

                        true
                      ]
                    }
                  ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],

"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "endpoint": {
                    "url": {
                      "ref": "Endpoint"
                    },
                    "properties": {},
                    "headers": {}
```

```
                    },
                    "type": "endpoint"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [

  {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "type": "tree",

"rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },
                                        true
                                    ]
                                },
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
```

```
                              {
                                "ref": "UseDualStack"

                              },
                              true
                            ]
                        }
                    ],
                  "type": "tree",
                  "rules": [
                      {
                          "conditions": [
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                      "fn": "getAttr",

"argv": [

                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsFIPS"
                                        ]
                                    }
                                ]
                            },
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {

"fn": "getAttr",

                                        "argv": [
                                          {
                                              "ref": "PartitionResult"
                                          },
                                          "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {
```

```
          "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                              },
                              "type": "endpoint"
                          }
                        ]
                    },
                    {
                      "conditions": [],
                      "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                        "type": "error"
                    }
                  ]
                },
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }
                  ],
                  "type": "tree",

        "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsFIPS"
```

```
                              ]
                                }
                              ]
                            }
                          ],
                          "type": "tree",
                          "rules": [
                            {
                              "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                  "properties": {},
                                  "headers": {}

              },

                                  "type": "endpoint"
                              }
                            ]
                          },
                          {
                            "conditions": [],
                            "error": "FIPS is enabled but this partition does not support FIPS",
                            "type": "error"
                          }
                        ]
                      },
                      {
                        "conditions": [
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              {
                                "ref": "UseDualStack"
                              },
                              true
                            ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
                          {
                            "conditions": [
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
```

```
                                {
                             "fn": "getAttr",
                                "argv": [
                                  {
                                      "ref": "PartitionResult"
                                  },
                                  "supportsDualStack"
                                ]
                              }
                            ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
                          {

                              "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                              },
                              "type": "endpoint"
                          }
                        ]
                      },
                      {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does
 not support DualStack",
                        "type": "error"
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ]
              }
```

```
                        ]
                },
                {
                    "conditions":
[],
                    "error": "Invalid Configuration: Missing Region",
                    "type": "error"
                }
            ]
        },
        "smithy.rules#endpointTests": {
            "testCases": [
                {
                    "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                        }
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": true,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                        }
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                        }
                    },
```

```
        "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                }
            },

    "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
```

```
          {
              "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                }
              },
              "params": {
                "Region": "cn-north-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
          },
          {
              "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                }
              },
              "params": {
                "Region": "cn-north-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
          },
          {
              "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                }
              },

      "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
          },
          {
              "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                }
              },
```

          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },

      "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect":

```
{
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                    "error": "DualStack is enabled but this partition does not support DualStack"
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack":
true
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                  "expect": {
                    "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
                  },
                  "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                  }
                },
```

```
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
```

```
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {

"documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "Missing region",
          "expect": {
            "error": "Invalid Configuration: Missing Region"
          }
        }
```

```
                ],
                "version": "1.0"
              }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
             },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated":
{
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
```

```
              "items": "IdentityProviderSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
          "type": "structure",
          "members": {
            "IdentityProviderSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
                "smithy.api#required": {}
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
          },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
         },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
```

```
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
            "type":
"structure",
            "members": {
                "InstanceSummaries": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                    "traits": {
                        "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
```

```json
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "ProductUserSummaries"
            }
          }
        },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
```

```
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {

            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n    more sets of key-value pairs that you specify.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions operation.</p>"
          }
        },

    "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
```

```
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
      },
      "smithy.api#paginated":
{
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
```

```
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
```

```
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
            "type": "structure",
            "members": {

                "Username": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Product": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                        "smithy.api#required": {}
                    }
                },
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                        "smithy.api#required": {}
                    }

                },
                "Status": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The status of a product for a user.</p>",
                        "smithy.api#required": {}
                    }
                },
                "StatusMessage": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
                    }
                },
                "Domain": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The domain name of the user.</p>"
                    }
```

```
            },
            "SubscriptionStartDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
               }
            },
            "SubscriptionEndDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The end date of a subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Settings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n     subnets to provision VPC endpoints.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
```

```
          "IdentityProviderSummary":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
        "type": "string",

    "traits": {
          "smithy.api#length": {
            "min": 5,
            "max": 200
          },
          "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
```

```
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n      your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n      subnets to provision VPC endpoints, and the security group ID that
 is associated with the VPC\n      endpoints. The security group should permit inbound TCP port 1688
communication from resources\n      in the VPC.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related costs
will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
 For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n        monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/StartProductSubscription"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required":
{}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
```

```
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation":
  "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {

            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
  request.</p>",
          "smithy.api#error": "client"
        }
      },
```

```
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity\n    provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
```

        },
        "UpdateSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n   update any combination of settings in a single operation such as the:</p>\n
\n   \n   <p>Subnets which you want to add to provision VPC endpoints.</p>\n   \n
    \n   <p>Subnets which you want to remove the VPC endpoints from.</p>\n   \n
\n   <p>Security group ID which permits traffic to the VPC endpoints.</p>\n   \n   ",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n   require connectivity to activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
          }
        },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID

```
                that allows inbound TCP port 1688 communication between resources in\n      your VPC and the VPC endpoints for
activation servers.</p>"
                  }
                }
              },
              "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as\n      the subnets to provision VPC endpoints.</p>"
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
              "type": "structure",
              "members": {
                "message": {
                  "target": "smithy.api#String"
                }
              },
              "traits": {
                "smithy.api#documentation": "<p>A parameter is not valid.</p>",
                "smithy.api#error": "client"
              }
            }
          }
        }
      }
      {
        "smithy": "2.0",
        "metadata": {
          "suppressions": [
            {
              "id": "HttpMethodSemantics",
              "namespace": "*"
            },
            {
              "id": "HttpResponseCodeSemantics",
              "namespace": "*"
            },
            {
              "id": "PaginatedTrait",
              "namespace": "*"
            },
            {
              "id": "HttpHeaderTrait",
              "namespace": "*"
            },
            {
              "id": "HttpUriConflict",
              "namespace": "*"
            },
```

```
            {
                "id": "Service",
                "namespace": "*"
            }
        ]
    },
    "shapes": {
        "com.amazonaws.licensemanager#AWSLicenseManager": {
            "type": "service",
            "version": "2018-08-01",
            "operations": [
                {
                    "target": "com.amazonaws.licensemanager#AcceptGrant"
                },


{
                    "target": "com.amazonaws.licensemanager#CheckInLicense"
                },
                {
                    "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
                },
                {
                    "target": "com.amazonaws.licensemanager#CheckoutLicense"
                },
                {
                    "target": "com.amazonaws.licensemanager#CreateGrant"
                },
                {
                    "target": "com.amazonaws.licensemanager#CreateGrantVersion"
                },
                {
                    "target": "com.amazonaws.licensemanager#CreateLicense"
                },
                {
                    "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
                },
                {
                    "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
                },
                {
                    "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
                },
                {
                    "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
                },
                {
                    "target": "com.amazonaws.licensemanager#CreateToken"
                },
```

```
    {
      "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
      "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    }

},
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
    {
      "target": "com.amazonaws.licensemanager#GetGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseUsage"
    },
    {
      "target": "com.amazonaws.licensemanager#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
     },
    {
      "target": "com.amazonaws.licensemanager#ListDistributedGrants"
```

```
        },
        {
            "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
        },
        {
            "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
        },
        {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
        },
        {
            "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
        },
        {
            "target": "com.amazonaws.licensemanager#ListLicenses"
        },
        {
            "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
        },
        {

            "target": "com.amazonaws.licensemanager#ListLicenseVersions"
        },
        {
            "target": "com.amazonaws.licensemanager#ListReceivedGrants"
        },
        {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
        },
        {
            "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
        },
        {
            "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
        },
        {
            "target": "com.amazonaws.licensemanager#ListResourceInventory"
        },
        {
            "target": "com.amazonaws.licensemanager#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanager#ListTokens"
        },
        {
            "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
        },
        {
```

          "target": "com.amazonaws.licensemanager#RejectGrant"
        },
        {
          "target": "com.amazonaws.licensemanager#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanager#UntagResource"
        },
        {
          "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
        },
        {
          "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
        },
        {
          "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
        },
        {
          "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager",
          "arnNamespace": "license-manager",
          "cloudFormationName": "LicenseManager",
          "cloudTrailEventSource": "licensemanager.amazonaws.com",
          "endpointPrefix": "license-manager"
        },
        "aws.auth#sigv4": {
          "name": "license-manager"
        },
        "aws.protocols#awsJson1_1": {},
        "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software vendors across multiple \n   Amazon Web Services accounts and on-premises servers.</p>",
        "smithy.api#title": "AWS License Manager",
        "smithy.api#xmlNamespace": {
          "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
        },
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },

```
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "Boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
    "type": "Boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "String"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]

            }
        ],
        "type": "tree",
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
```

```
                    ]
                }
            ],
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
            "type": "error"
        },
        {

"conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
            "type": "error"
        },
        {
            "conditions": [],
            "endpoint": {
                "url": {
                    "ref": "Endpoint"
                },
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ]
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ],
    "type": "tree",
    "rules": [
```

```
{
    "conditions": [
        {
            "fn": "aws.partition",
            "argv": [
                {
                    "ref": "Region"
                }
            ],
            "assign": "PartitionResult"
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref":
"UseFIPS"
                        },
                        true
                    ]
                },
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
```

```
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              }
            ]
          },
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsDualStack"
                ]
              }
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
              "properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ]
      },
      {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both",
        "type": "error"
      }
    ]
  },
```

```
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        },
```

```
                                  {
                                      "conditions": [],
                                      "error": "FIPS is enabled but this
partition does not support FIPS",
                                      "type": "error"
                                  }
                              ]
                          },
                          {
                              "conditions": [
                                  {
                                      "fn": "booleanEquals",
                                      "argv": [
                                          {
                                              "ref": "UseDualStack"
                                          },
                                          true
                                      ]
                                  }
                              ],
                              "type": "tree",
                              "rules": [
                                  {
                                      "conditions": [
                                          {
                                              "fn": "booleanEquals",
                                              "argv": [
                                                  true,
                                                  {
                                                      "fn": "getAttr",
                                                      "argv": [
                                                          {
                                                              "ref": "PartitionResult"
                                                          },

"supportsDualStack"
                                                      ]
                                                  }
                                              ]
                                          }
                                      ],
                                      "type": "tree",
                                      "rules": [
                                          {
                                              "conditions": [],
                                              "endpoint": {
                                                  "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
```

```
                              "properties": { },
                                "headers": { }


                  },
                        "type": "endpoint"
                      }
                  ]
                },
                {
                    "conditions": [],
                    "error": "DualStack is enabled but this partition does not support DualStack",
                    "type": "error"
                  }
              ]
          },
          {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": { },
                  "headers": { }
                },
                "type": "endpoint"
            }
        ]
      }
  ]
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
      {
          "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint":
{
                "url": "https://license-manager.af-south-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "af-south-1",
              "UseFIPS": false,
```

```
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-northeast-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-northeast-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-3.amazonaws.com"
          }
```

```
          },
          "params": {
            "Region": "ap-northeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }

        },
        {
          "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-1.amazonaws.com"

}
          },
          "params": {
            "Region": "ap-southeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
```

```
          {
  "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-3.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ca-central-1.amazonaws.com"
            }
          },

         "params": {
            "Region": "ca-central-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-central-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-central-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-north-1.amazonaws.com"
            }
```

```
          },
          "params": {
            "Region": "eu-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region":
"eu-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-west-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
          "expect": {

            "endpoint": {
              "url": "https://license-manager.eu-west-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-west-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
```

```
{
    "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.eu-west-3.amazonaws.com"
        }
    },
    "params": {
        "Region": "eu-west-3",
        "UseFIPS": false,

"UseDualStack": false
    }
},
{
    "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.me-south-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "me-south-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.sa-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "sa-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-east-1.amazonaws.com"
        }
    },
    "params": {
```

```
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }

},
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-east-2.amazonaws.com"
          }

        },
        "params": {
          "Region": "us-east-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-east-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-2",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region
```

us-west-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager.us-west-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-west-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-fips.us-west-1.amazonaws.com"
                  }
                },
                "params": {

          "Region": "us-west-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager.us-west-2.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-west-2",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

"expect": {
                  "endpoint": {
                    "url": "https://license-manager-fips.us-west-2.amazonaws.com"
                  }
                },
                "params": {

```
        "Region": "us-west-2",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,

        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
```

```
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
                }
              },
              "params": {
                "Region": "cn-northwest-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
```

```
            "UseDualStack": true
          }
        },

    {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-gov-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-west-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
```

```
                    "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
                  }
              },
              "params": {
                 "Region": "us-gov-west-1",
                 "UseFIPS": true,
                 "UseDualStack": false
              }
          },
          {
              "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                 "endpoint": {
                    "url": "https://license-manager-fips.us-gov-east-1.api.aws"
                 }
              },

      "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                 }
             },
             {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                   "endpoint": {
                      "url": "https://license-manager.us-gov-east-1.api.aws"
                   }
                },
                "params": {
                   "Region": "us-gov-east-1",
                   "UseFIPS": false,
                   "UseDualStack": true
                }
             },
             {
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
                "expect": {
                   "endpoint": {
                      "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
                   }
                },
                "params": {
                   "Region": "us-iso-east-1",
                   "UseFIPS": false,
                   "UseDualStack": false
```

```
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": true,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
                        }
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                        "error": "DualStack is enabled
but this partition does not support DualStack"
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
                        }
                    },
                    "params": {
                        "Region": "us-isob-east-1",
                        "UseFIPS": false,
```

```
            "UseDualStack": false
          }
        },

        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
        },

  "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

  "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
```

```
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,

    "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
```

"documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanager#AcceptGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

},

    "output": {
        "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"

}
    ],
    "traits": {
        "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
    }
},
"com.amazonaws.licensemanager#AcceptGrantRequest": {

```
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant
ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#AccessDeniedException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
```

```json
        "traits": {
            "aws.protocols#awsQueryError": {

"code": "ServiceAccessDenied",
                "httpResponseCode": 401
            },
            "smithy.api#documentation": "<p>Access to resource denied.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 401
        }
    },
    "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
        "type": "enum",
        "members": {
            "DISTRIBUTED_GRANTS_ONLY": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
                }
            },
            "ALL_GRANTS_PERMITTED_BY_ISSUER": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#AllowedOperation": {

"type": "enum",
        "members": {
            "CREATE_GRANT": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CreateGrant"
                }
            },
            "CHECKOUT_LICENSE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CheckoutLicense"
                }
            },
            "CHECKOUT_BORROW_LICENSE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CheckoutBorrowLicense"
```

```
          }
        },
        "CHECK_IN_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckInLicense"
          }
        },
        "EXTEND_CONSUMPTION_LICENSE":
{
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ExtendConsumptionLicense"
          }
        },
        "LIST_PURCHASED_LICENSES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ListPurchasedLicenses"
          }
        },
        "CREATE_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateToken"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#AllowedOperationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#AllowedOperation"
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max":
7
        }
      }
    },
    "com.amazonaws.licensemanager#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 2048
```

```
        },
        "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-
z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
      }
    },
    "com.amazonaws.licensemanager#ArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
      }
    },
    "com.amazonaws.licensemanager#AuthorizationException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError":
{
          "code": "AuthorizationFailure",
          "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n         policy associated with this account.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 403
      }
    },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
      "type": "structure",
      "members": {
        "LastRunTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes automated discovery.</p>"
      }
    },

  "com.amazonaws.licensemanager#Boolean": {
      "type": "boolean",
      "traits": {
```

```
          "smithy.api#default": false
        }
      },
      "com.amazonaws.licensemanager#BorrowConfiguration": {
        "type": "structure",
        "members": {
          "AllowEarlyCheckIn": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxTimeToLiveInMinutes": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
              "smithy.api#required": {}
            }
          }

        },
        "traits": {
          "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#BoxBoolean": {
        "type": "boolean"
      },
      "com.amazonaws.licensemanager#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanager#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanager#CheckInLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanager#AuthorizationException"

        },
            {
            "target": "com.amazonaws.licensemanager#ConflictException"
        },
            {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
            {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
            {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
            {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
            {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
        }
    },
    "com.amazonaws.licensemanager#CheckInLicenseRequest":
{
        "type": "structure",
        "members": {
            "LicenseConsumptionToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License consumption token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Beneficiary": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License beneficiary.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
```

```
    },
    "com.amazonaws.licensemanager#CheckInLicenseResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
          "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
        "type": "structure",
        "members":
  {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
              "smithy.api#required": {}
            }
          },
          "DigitalSignatureMethod": {
            "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
            "traits": {
              "smithy.api#documentation": "<p>Digital signature method. The possible
 value is JSON Web Signature (JWS) algorithm PS384.\n          For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
              "smithy.api#required": {}
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          },
          "CheckoutMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about constraints.</p>"
            }
```

```
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },
        "EntitlementsAllowed":
{
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "SignedToken": {
          "target": "com.amazonaws.licensemanager#SignedToken",
          "traits": {
            "smithy.api#documentation": "<p>Signed token.</p>"
          }
        },
```

```
        "IssuedAt": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
          }
        },
        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
```

```json
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n          <p>If the
account that created the license is the same that is performing the check out, you must\n          specify the account
as the beneficiary.</p>\n        </note>"
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
      "type": "structure",
      "members": {
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product
SKU.</p>",
            "smithy.api#required": {}
          }
        },
        "CheckoutType": {
          "target": "com.amazonaws.licensemanager#CheckoutType",
          "traits": {
            "smithy.api#documentation": "<p>Checkout type.</p>",
            "smithy.api#required": {}
          }
        },
        "KeyFingerprint": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
            "smithy.api#required": {}
          }
        },
```

```
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        }
      },
      "traits": {

    "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
      "type": "structure",
      "members": {
        "CheckoutType": {
          "target": "com.amazonaws.licensemanager#CheckoutType",
          "traits": {
            "smithy.api#documentation": "<p>Checkout type.</p>"
          }
        },
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
```

```
      },
      "EntitlementsAllowed": {
        "target": "com.amazonaws.licensemanager#EntitlementDataList",
        "traits": {
          "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
        }
      },
       "SignedToken": {
        "target": "com.amazonaws.licensemanager#SignedToken",
        "traits": {
          "smithy.api#documentation": "<p>Signed token.</p>"
        }
      },
      "NodeId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Node ID.</p>"
        }
      },
      "IssuedAt": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
        }
      },
      "Expiration": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
        }
      },
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#CheckoutType": {
  "type": "enum",
  "members": {
    "PROVISIONAL": {
      "target": "smithy.api#Unit",
```

```
        "traits": {
            "smithy.api#enumValue": "PROVISIONAL"
        }
    },
    "PERPETUAL": {
        "target": "smithy.api#Unit",
        "traits": {
            "smithy.api#enumValue": "PERPETUAL"
        }
    }
    }
},
"com.amazonaws.licensemanager#ClientRequestToken": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 36
        }
    }
},
"com.amazonaws.licensemanager#ClientToken": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 2048
        },
        "smithy.api#pattern": "^\\S+$"
    }
},
"com.amazonaws.licensemanager#ConflictException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
    },
    "traits": {
        "aws.protocols#awsQueryError": {
            "code": "ConflictException",
            "httpResponseCode":
409
        },
        "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 409
```

```
            }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
            "type": "structure",
            "members": {
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#ResourceType",
                    "traits": {
                        "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
                    }
                },
                "ConsumedLicenses": {
                    "target": "com.amazonaws.licensemanager#BoxLong",
                    "traits": {
                        "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
                    }
                }
            },

    "traits": {
            "smithy.api#documentation": "<p>Details about license consumption.</p>"
        }
    },
        "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
            }
        },
        "com.amazonaws.licensemanager#ConsumptionConfiguration": {
            "type": "structure",
            "members": {
                "RenewType": {
                    "target": "com.amazonaws.licensemanager#RenewType",
                    "traits": {
                        "smithy.api#documentation": "<p>Renewal frequency.</p>"
                    }
                },
                "ProvisionalConfiguration": {
                    "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
                    }
                },
                "BorrowConfiguration": {
                    "target": "com.amazonaws.licensemanager#BorrowConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
```

```
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target":
"com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrantRequest": {
```

```
    "type": "structure",
    "members": {
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientToken",
        "traits": {
          "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant name.</p>",
          "smithy.api#required": {}

  }
      },
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
          "smithy.api#required": {}
        }
      },
      "Principals": {
        "target": "com.amazonaws.licensemanager#PrincipalArnList",
        "traits": {
          "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n      (ARN):</p>\n    <ul>\n      <li>\n         <p>An Amazon Web Services
account, which includes only the account specified.</p>\n      </li>\n    </ul>\n    <ul>\n      <li>\n
   <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n         </li>\n      </ul>\n
<ul>\n
      <li>\n         <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n      </ul>",
          "smithy.api#required": {}
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
        "traits": {
```

```
            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },

    "com.amazonaws.licensemanager#CreateGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersion":
    {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
    }
  },
  "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
    "type": "structure",
    "members": {
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientToken",
        "traits": {
          "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

  "smithy.api#required": {}
        }
      },
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
```

                "smithy.api#documentation": "<p>Grant name.</p>"
              }
            },
            "AllowedOperations": {
              "target": "com.amazonaws.licensemanager#AllowedOperationList",
              "traits": {
                "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanager#GrantStatus",
              "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
              }
            },
            "StatusReason": {
              "target": "com.amazonaws.licensemanager#StatusReasonMessage",
              "traits": {
                "smithy.api#documentation": "<p>Grant status reason.</p>"
              }
            },
            "SourceVersion": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Current version of the grant.</p>"
              }
            },
            "Options": {
              "target": "com.amazonaws.licensemanager#Options",
              "traits": {
                "smithy.api#documentation": "<p>The options specified for the grant.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {

```
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>New version of the grant.</p>"
                    }
                }
            },
            "traits":
{
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicense": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RedirectException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Creates a license.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of
licenses purchased and used.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Name of the license configuration.</p>",
```

```
                "smithy.api#required": { }
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Description of the license configuration.</p>"
              }
            },
            "LicenseCountingType": {
              "target": "com.amazonaws.licensemanager#LicenseCountingType",
               "traits": {
                "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
                "smithy.api#required": { }
              }
            },
            "LicenseCount": {
              "target": "com.amazonaws.licensemanager#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
              }
            },
            "LicenseCountHardLimit": {
              "target": "com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n        blocks the launch of new instances.</p>"
              }
            },
            "LicenseRules": {
               "target":
 "com.amazonaws.licensemanager#StringList",
               "traits": {
                 "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n        vary by dimension, as follows.</p>\n
<ul>\n        <li>\n          <p>\n            <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
       <code>licenseAffinityToHost</code> |\n            <code>maximumCores</code> |
<code>minimumCores</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n          <code>maximumCores</code> |
<code>minimumCores</code> |\n          <code>maximumSockets</code> | <code>minimumSockets</code> |\n
       <code>maximumVcpus</code> | <code>minimumVcpus</code>\n          </p>\n        </li>\n
<li>\n
  <p>\n            <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n          <code>maximumSockets</code> |
<code>minimumSockets</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n          <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n          </p>\n        </li>\n        </ul>\n        <p>The unit for
```

<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for <code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n <code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n <code>True</code> and <code>False</code>.</p>"

```
                    }
                },

  "Tags": {
                    "target": "com.amazonaws.licensemanager#TagList",
                    "traits": {
                        "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
                    }
                },
                "DisassociateWhenNotFound": {
                    "target": "com.amazonaws.licensemanager#BoxBoolean",
                    "traits": {
                        "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                    }
                },
                "ProductInformationList": {
                    "target": "com.amazonaws.licensemanager#ProductInformationList",
                    "traits": {
                        "smithy.api#documentation": "<p>Product information.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
            "type": "structure",
            "members": {
                "LicenseConfigurationArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
            "type": "operation",
```

```
"input": {
    "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
},
"output": {
    "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
},
"errors": [
    {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
        "target": "com.amazonaws.licensemanager#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
}
},
"com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {

            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
                "smithy.api#required": {}
            }
        },
        "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
                "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n        For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
```

```
the <i>License Manager User Guide</i>.</p>",
                    "smithy.api#required": {}
                }
            },
            "DestinationLicenseContext":
{
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

                "traits": {
                    "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
```

```
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "ReportGeneratorName":
{
          "target": "com.amazonaws.licensemanager#ReportGeneratorName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>",
            "smithy.api#required": {}
          }
        },
        "Type": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n       <ul>\n          <li>\n             <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n          </li>\n          <li>\n             <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n          </li>\n
</ul>",
            "smithy.api#required": {}

    }
        },
```

```
    "ReportContext": {
      "target": "com.amazonaws.licensemanager#ReportContext",
      "traits": {
        "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
        "smithy.api#required": {}
      }
    },
    "ReportFrequency": {
      "target": "com.amazonaws.licensemanager#ReportFrequency",
      "traits": {
        "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
        "smithy.api#required": {}
      }
    },
    "ClientToken": {
      "target": "com.amazonaws.licensemanager#ClientRequestToken",
      "traits": {
        "smithy.api#documentation": "<p>Unique,
 case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
        "smithy.api#required": {}
      }
    },
    "Description": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Description of the report generator.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanager#TagList",
      "traits": {
        "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
  "type": "structure",
  "members": {
    "LicenseManagerReportGeneratorArn":
{
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
```

```
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#CreateLicenseRequest": {
            "type": "structure",
            "members": {
              "LicenseName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>License name.</p>",
                  "smithy.api#required": {}
                }
              },
              "ProductName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Product
name.</p>",
                  "smithy.api#required": {}
                }
              },
              "ProductSKU": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Product SKU.</p>",
                  "smithy.api#required": {}
                }
              },
              "Issuer": {
                "target": "com.amazonaws.licensemanager#Issuer",
                "traits": {
                  "smithy.api#documentation": "<p>License issuer.</p>",
                  "smithy.api#required": {}
                }
              },
              "HomeRegion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Home Region for the license.</p>",
                  "smithy.api#required": {}
```

```
            }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
              "smithy.api#required": {}
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>",
              "smithy.api#required": {}
            }
          },
          "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license.</p>"
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          }
        },
```

```
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersion": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target":
"com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>",
            "smithy.api#required": {}
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#Issuer",
```

              "traits":
{
                "smithy.api#documentation": "<p>License issuer.</p>",
                "smithy.api#required": {}
              }
            },
            "HomeRegion": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Home Region of the license.</p>",
                "smithy.api#required": {}
              }
            },
            "Validity": {
              "target": "com.amazonaws.licensemanager#DatetimeRange",
              "traits": {
                "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
                "smithy.api#required": {}
              }
            },
            "LicenseMetadata": {
              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                "smithy.api#documentation": "<p>Information about the license.</p>"
              }
            },
            "Entitlements": {
              "target": "com.amazonaws.licensemanager#EntitlementList",
              "traits": {
                "smithy.api#documentation": "<p>License entitlements.</p>",
                "smithy.api#required": {}
              }
            },
            "ConsumptionConfiguration": {
              "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
              "traits": {
                "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n   running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n   usage.</p>",
                "smithy.api#required": {}
              }
            },

      "Status": {
              "target": "com.amazonaws.licensemanager#LicenseStatus",
              "traits": {
                "smithy.api#documentation": "<p>License status.</p>",

```
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the license.</p>"
            }
          }
        },
        "traits": {

  "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New version of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
```

```
{}
        }
    },
    "com.amazonaws.licensemanager#CreateToken": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateTokenRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateTokenResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },

            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateTokenRequest": {
        "type": "structure",
        "members": {
            "LicenseArn": {
```

```
                "target": "com.amazonaws.licensemanager#Arn",
                "traits":
{
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n         JWT token.</p>",
                    "smithy.api#required": {}
                }
            },
            "RoleArns": {
                "target": "com.amazonaws.licensemanager#ArnList",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n         License Manager does not check whether the roles are in use.</p>"
                }
            },
            "ExpirationInDays": {
                "target": "com.amazonaws.licensemanager#Integer",
                "traits": {
                    "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
                }
            },
            "TokenProperties": {
                "target": "com.amazonaws.licensemanager#MaxSize3StringList",
                "traits": {
                    "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n         to the amr claim of the JWT token.</p>"
                }
            },
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateTokenResponse": {
        "type": "structure",
        "members": {
            "TokenId": {
                "target": "com.amazonaws.licensemanager#String",
```

```
        "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
        }
    },
    "TokenType": {
        "target": "com.amazonaws.licensemanager#TokenType",
        "traits": {
            "smithy.api#documentation": "<p>Token type.</p>"
        }
    },
    "Token": {
        "target": "com.amazonaws.licensemanager#TokenString",
        "traits": {
            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
        }
    }
},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#DateTime": {
    "type": "timestamp"
},
"com.amazonaws.licensemanager#DatetimeRange": {
    "type": "structure",
    "members": {
        "Begin": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>Start of the time range.</p>",
                "smithy.api#required": {}
            }
        },
        "End": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>End of the time range.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
    }
},
"com.amazonaws.licensemanager#DeleteGrant": {
    "type": "operation",
    "input": {
```

```
                    "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
                },
                "output": {
                    "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
                },
                "errors": [
                    {
                        "target": "com.amazonaws.licensemanager#AccessDeniedException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#AuthorizationException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#ServerInternalException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#ValidationException"
                    }
                ],
                "traits": {
                    "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
                }
            },
            "com.amazonaws.licensemanager#DeleteGrantRequest": {
                "type": "structure",
                "members": {
                    "GrantArn": {
                        "target": "com.amazonaws.licensemanager#Arn",
                        "traits": {
                            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                            "smithy.api#required": {}
                        }
                    },
                    "StatusReason": {
                        "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                        "traits": {
                            "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
                        }
```

```
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the
grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation":
"<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
```

        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },

```
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
          "type": "structure",
          "members":
{
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>ID of the license configuration.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
          },
          "errors": [
            {
```

```
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
          "type": "structure",
          "members": {
            "LicenseManagerReportGeneratorArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {

          "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
```

```
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "SourceVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the license.</p>",
            "smithy.api#required":
{}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseResponse": {
      "type": "structure",
      "members": {
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "DeletionDate": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
```

```json
      },
      "com.amazonaws.licensemanager#DeleteToken": {
        "type": "operation",
         "input": {
          "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },

      {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteTokenRequest": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>",
              "smithy.api#required": {}
            }
          }
```

```
        },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}


      }
    },
    "com.amazonaws.licensemanager#DigitalSignatureMethod": {
      "type": "enum",
      "members": {
        "JWT_PS384": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "JWT_PS384"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#Entitlement": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement name.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#Long",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
          }
        },
        "Overage": {
```

```
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement unit.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowCheckIn": {
            "target":
"com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#EntitlementData": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement data name.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement
data value.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement data unit.</p>",
              "smithy.api#required": {}
            }
          }
```

```
          },
        "traits": {
          "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#EntitlementDataList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#EntitlementData"
        }
      },
      "com.amazonaws.licensemanager#EntitlementDataUnit": {
        "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {

"smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "SECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Seconds"
            }
          },
          "MICROSECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Microseconds"
            }
          },
          "MILLISECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Milliseconds"
            }
          },
          "BYTES": {
            "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS":
{
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits"
      }
    },
    "KILOBITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobits"
      }
    },
    "MEGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits"
      }
    },
    "GIGABITS": {
```

```
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Gigabits"
            }
         },
         "TERABITS": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Terabits"

}
         },
         "PERCENT": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Percent"
            }
         },
         "BYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Bytes/Second"
            }
         },
         "KILOBYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Kilobytes/Second"
            }
         },
         "MEGABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Megabytes/Second"
            }
         },
         "GIGABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
             "traits": {
               "smithy.api#enumValue": "Gigabytes/Second"
            }
         },
         "TERABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Terabytes/Second"
            }
         },
```

```
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
          }
        },
        "TERABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits/Second"
          }
        },
        "COUNT_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count/Second"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Entitlement"
      }
    },
    "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
      "type": "structure",
```

        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#EntitlementUnit": {
        "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "SECONDS":
{
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Seconds"
            }
          },
          "MICROSECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Microseconds"
            }
          },
          "MILLISECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Milliseconds"
            }
          },
          "BYTES": {
            "target": "smithy.api#Unit",
            "traits": {

```
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits"
      }
    },
    "KILOBITS": {
      "target": "smithy.api#Unit",
      "traits": {

        "smithy.api#enumValue": "Kilobits"
      }
    },
    "MEGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits"
      }
    },
    "GIGABITS": {
      "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue": "Gigabits"
      }
    },
    "TERABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabits"
      }
    },
    "PERCENT": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Percent"
      }
    },
    "BYTES_PER_SECOND": {
      "target":
"smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes/Second"
      }
    },
    "KILOBYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes/Second"
      }
    },
    "MEGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes/Second"
      }
    },
    "GIGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes/Second"
      }
    },
    "TERABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {

"smithy.api#enumValue": "Terabytes/Second"
      }
    },
```

```
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
          }
        },

    "TERABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits/Second"
          }
        },
        "COUNT_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count/Second"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsage": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage name.</p>",
            "smithy.api#required": {}
          }
```

```
            },
            "ConsumedValue": {
                "target": "com.amazonaws.licensemanager#String",

            "traits": {
                    "smithy.api#documentation": "<p>Resource usage consumed.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxCount": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
                }
            },
            "Unit": {
                "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
                "traits": {
                    "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
        }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList":
{
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#EntitlementUsage"
        }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
    }
  },
  "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
    "type": "structure",
    "members": {
      "LicenseConsumptionToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License consumption token.</p>",
          "smithy.api#required": {}

    }
      },
      "DryRun": {
        "target": "com.amazonaws.licensemanager#Boolean",
        "traits": {
          "smithy.api#default": false,
          "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
    "type": "structure",
    "members": {
```

```
      "LicenseConsumptionToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License consumption token.</p>"
        }
      },

      "Expiration": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#FailedDependencyException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      },
      "ErrorCode": {
        "target": "com.amazonaws.licensemanager#String"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "FailedDependency",
        "httpResponseCode": 424
      },
      "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 424
    }
  },
  "com.amazonaws.licensemanager#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#FilterName",
        "traits": {
          "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
        }
      },
```

```
        "Values": {
          "target": "com.amazonaws.licensemanager#FilterValues",
          "traits": {
            "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n        describe operation. Filters can be used
to match a set of resources by specific criteria,\n        such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanager#FilterLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "FilterLimitExceeded",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Filter"
      }
    },
    "com.amazonaws.licensemanager#FilterName": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#FilterValue": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#FilterValues": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#FilterValue",
        "traits": {
```

```
            "smithy.api#xmlName": "item"
          }
        }
      },
      "com.amazonaws.licensemanager#Filters": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Filter",
          "traits": {
            "smithy.api#xmlName": "item"
          }
        }
      },
      "com.amazonaws.licensemanager#GetAccessToken": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
tokens\n        are valid for one hour.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetAccessTokenRequest": {
        "type": "structure",
        "members": {
          "Token": {
```

```
                "target": "com.amazonaws.licensemanager#TokenString",
                "traits": {
                  "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
                  "smithy.api#required": {}
                }
              }
            },
            "TokenProperties": {
              "target": "com.amazonaws.licensemanager#MaxSize3StringList",
              "traits": {
                "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
          "type": "structure",
          "members": {
            "AccessToken": {
              "target": "com.amazonaws.licensemanager#TokenString",
              "traits": {
                "smithy.api#documentation": "<p>Temporary access token.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#GetGrant": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#GetGrantRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#GetGrantResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
    },
    {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
    },
    {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
        "target": "com.amazonaws.licensemanager#ValidationException"
    }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
    }
},
"com.amazonaws.licensemanager#GetGrantRequest": {
    "type": "structure",
    "members": {
        "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
            }
        },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#GetGrantResponse": {
    "type": "structure",
    "members": {
        "Grant": {
            "target": "com.amazonaws.licensemanager#Grant",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>Grant details.</p>"
                }
            }
        },
        "traits": {

"smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#GetLicense": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
    }
},
"com.amazonaws.licensemanager#GetLicenseConfiguration": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
    },
    "output": {
```

```
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits":
{
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurationId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
```

```
        }
      },
      "LicenseConfigurationArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
        }
      },
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Name of the license configuration.</p>"
        }
      },
      "Description": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Description of the license configuration.</p>"
        }
      },
      "LicenseCountingType": {
        "target": "com.amazonaws.licensemanager#LicenseCountingType",
        "traits": {
          "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",

        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
```

```
            "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License configuration status.</p>"
          }

      },
          "OwnerAccountId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
            }
          },
          "ConsumedLicenseSummaryList": {
            "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
            }
          },
          "ManagedResourceSummaryList": {
            "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
             "traits": {
              "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
            }
          },
          "ProductInformationList": {
            "target": "com.amazonaws.licensemanager#ProductInformationList",
            "traits": {
              "smithy.api#documentation": "<p>Product information.</p>"
            }
          },
          "AutomatedDiscoveryInformation": {
            "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
            "traits": {
              "smithy.api#documentation": "<p>Automated discovery information.</p>"
            }
          },
          "DisassociateWhenNotFound": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
```

```
                        "traits": {
                            "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
                        }
                    }
                },
                "traits": {
                    "smithy.api#output": {}
                }
            },
            "com.amazonaws.licensemanager#GetLicenseConversionTask": {
                "type": "operation",
                "input": {
                    "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
                },
                "output": {
                    "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
                },
                "errors": [
                    {
                        "target": "com.amazonaws.licensemanager#AccessDeniedException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#AuthorizationException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                    },
                    {

"target": "com.amazonaws.licensemanager#RateLimitExceededException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#ServerInternalException"
                    }
                ],
                "traits": {
                    "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
                }
            },
            "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
                "type": "structure",
                "members": {
                    "LicenseConversionTaskId": {
                        "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                        "traits": {
                            "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
```

```
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                "traits": {
                    "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
                }
            },
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
                }
            },
            "SourceLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
                }
            },
            "DestinationLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
                }
```

```
                },

        "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                }
            },
            "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

            },
            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
            "LicenseManagerReportGeneratorArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
                    "smithy.api#required":
{}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {
            "ReportGenerator": {
                "target": "com.amazonaws.licensemanager#ReportGenerator",
                "traits": {
                    "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
```

```
        },
      "com.amazonaws.licensemanager#GetLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseResponse": {
        "type": "structure",
        "members": {
          "License": {
            "target": "com.amazonaws.licensemanager#License",
            "traits": {
              "smithy.api#documentation": "<p>License details.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },

    "com.amazonaws.licensemanager#GetLicenseUsage": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
        },
        "errors": [
          {
```

```
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {

  "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
      "type": "structure",
      "members": {
        "LicenseUsage": {
          "target":
"com.amazonaws.licensemanager#LicenseUsage",
          "traits": {
            "smithy.api#documentation": "<p>License usage details.</p>"
          }
        }
```

```
      },
      "traits": {
        "smithy.api#output": { }
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#input": { }
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"
```

```
            }
        },
        "SnsTopicArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
            }
        },
        "OrganizationConfiguration": {
            "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n        cross-account discovery.</p>"
            }
        },
        "EnableCrossAccountsDiscovery": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
                "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
            }

        },
        "LicenseManagerResourceShareArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n        provides member accounts with access to this share.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#Grant": {
    "type": "structure",
    "members": {
        "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
            }
        },
        "GrantName": {
```

```
"target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant name.</p>",
            "smithy.api#required": {}
          }
        },
        "ParentArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Parent ARN.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>License ARN.</p>",
            "smithy.api#required": {}
          }
        },
        "GranteePrincipalArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {

"smithy.api#documentation": "<p>The grantee principal ARN.</p>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantStatus": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Grant status reason.</p>"
          }
        },
```

```
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant version.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
        "traits": {
          "smithy.api#documentation": "<p>Granted operations.</p>",
          "smithy.api#required": {}
        }
      },
      "Options": {
        "target": "com.amazonaws.licensemanager#Options",
        "traits": {
          "smithy.api#documentation": "<p>The options specified for the grant.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>Describes a grant.</p>"
    }
  },
  "com.amazonaws.licensemanager#GrantList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Grant"
    }
  },
  "com.amazonaws.licensemanager#GrantStatus": {
    "type": "enum",
    "members": {
      "PENDING_WORKFLOW": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_WORKFLOW"
        }
      },
      "PENDING_ACCEPT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_ACCEPT"
        }
      },
      "REJECTED": {
```

```
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"

        }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits":
{
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
      "type": "structure",
      "members": {
```

```
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
      }
    },
    "LicenseName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>License name.</p>"
      }

    },
    "ProductName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product name.</p>"
      }
    },
    "ProductSKU": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product SKU.</p>"
      }
    },
    "Issuer": {
      "target": "com.amazonaws.licensemanager#IssuerDetails",
      "traits": {
        "smithy.api#documentation": "<p>Granted license issuer.</p>"
      }
    },
    "HomeRegion": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanager#LicenseStatus",
      "traits": {
        "smithy.api#documentation": "<p>Granted license status.</p>"
      }
    },
    "Validity": {
      "target": "com.amazonaws.licensemanager#DatetimeRange",
      "traits": {
        "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
```

```
ISO8601-UTC format.</p>"
              }
            },
        "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
            }
        },
        "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementList",
             "traits": {
                "smithy.api#documentation": "<p>License entitlements.</p>"
            }
        },
        "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
            }
        },
        "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
                "smithy.api#documentation": "<p>Granted license metadata.</p>"
            }
        },
        "CreateTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

    }
            },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Version of the granted license.</p>"
            }
        },
        "ReceivedMetadata": {
            "target": "com.amazonaws.licensemanager#ReceivedMetadata",
            "traits": {
                "smithy.api#documentation": "<p>Granted license received metadata.</p>"
            }
        }
    },
        "traits": {
```

```
          "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
        }
      },
      "com.amazonaws.licensemanager#GrantedLicenseList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#GrantedLicense"
        }
      },
      "com.amazonaws.licensemanager#ISO8601DateTime":
{
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-
1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
        }
      },
      "com.amazonaws.licensemanager#Integer": {
        "type": "integer"
      },
      "com.amazonaws.licensemanager#InvalidParameterValueException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidParameterValueProvided",
            "httpResponseCode": 400
          },
          "smithy.api#documentation":
"<p>One or more parameter values are not valid.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#InvalidResourceStateException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
```

```
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidResourceState",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },

      "com.amazonaws.licensemanager#InventoryFilter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the filter.</p>",
              "smithy.api#required": {}
            }
          },
          "Condition": {
            "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
            "traits": {
              "smithy.api#documentation": "<p>Condition of the filter.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }

      },
        "traits": {
          "smithy.api#documentation": "<p>An inventory filter.</p>"
        }
      },
      "com.amazonaws.licensemanager#InventoryFilterCondition": {
        "type": "enum",
        "members": {
          "EQUALS": {
            "target": "smithy.api#Unit",
```

```
          "traits": {
            "smithy.api#enumValue": "EQUALS"
          }
        },
        "NOT_EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "NOT_EQUALS"
          }
        },
        "BEGINS_WITH": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "BEGINS_WITH"
          }
        },
        "CONTAINS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CONTAINS"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#InventoryFilter"
      }
    },
    "com.amazonaws.licensemanager#Issuer": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Issuer name.</p>",
            "smithy.api#required": {}
          }
        },
        "SignKey": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
 of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
          }
        }
```

```
          },
        "traits": {
          "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#IssuerDetails": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>"
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#License": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

        "smithy.api#documentation": "<p>License name.</p>"
            }
```

```
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#IssuerDetails",
          "traits": {
            "smithy.api#documentation": "<p>License issuer.</p>"
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {

        "smithy.api#documentation": "<p>Home Region of the license.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        },
        "Entitlements": {
```

```
      "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
                "smithy.api#documentation": "<p>License entitlements.</p>"
            }
        },
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
          }
        },
        "LicenseMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>License metadata.</p>"
          }
        },
        "CreateTime": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>License
creation time.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfiguration": {
      "type": "structure",
      "members": {
        "LicenseConfigurationId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
          }
        },
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
           "traits": {
             "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
```

```
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules.</p>"
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
```

                    "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Status of the license configuration.</p>"
                }
            },
            "OwnerAccountId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

            }
            },
            "ConsumedLicenseSummaryList": {
                "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
                }
            },
            "ManagedResourceSummaryList": {
                "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
                }
            },
            "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                    "smithy.api#documentation": "<p>Product information.</p>"
                }
            },
            "AutomatedDiscoveryInformation": {

                "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
                "traits": {
                    "smithy.api#documentation": "<p>Automated discovery information.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license agreement that can be\n   consumed and enforced by License Manager. Components include specifications for the license\n   type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n   Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n   must be associated with a host), and the number of licenses purchased and used.</p>"

```
          }
        },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
        "type": "structure",
        "members": {
          "ResourceArn":
{
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
              }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of server resource.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
            }
          },
          "AssociationTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
            }
          },
          "AmiAssociationScope": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
```

```
            }
        },
        "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
            "type": "enum",
            "members":
{
            "AVAILABLE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "AVAILABLE"
                }
            },
            "DISABLED": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DISABLED"
                }
            }
        }
        },
        "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
            "type": "structure",
            "members": {
                "ResourceArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
                    }
                },
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#ResourceType",
                    "traits":
{
                        "smithy.api#documentation": "<p>Type of resource.</p>"
                    }
                },
                "ResourceStatus": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Status of the resource.</p>"
                    }
                },
                "ResourceOwnerId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
                    }
                },
```

```
              "AssociationTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                  "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
                }
              },

      "ConsumedLicenses": {
                "target": "com.amazonaws.licensemanager#BoxLong",
                "traits": {
                  "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
                }
              }
            },
          "traits": {
            "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
          }
        },
        "com.amazonaws.licensemanager#LicenseConfigurations": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#LicenseConfiguration"
          }
        },
        "com.amazonaws.licensemanager#LicenseConversionContext": {

    "type": "structure",
          "members": {
            "UsageOperation": {
              "target": "com.amazonaws.licensemanager#UsageOperation",
              "traits": {
                "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n          </p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {

          "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the conversion task.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "StartTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
```

```
            "traits": {
               "smithy.api#documentation":
  "<p>The time the conversion task was started at.</p>"
            }
         },
         "LicenseConversionTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
               "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
            }
         },
         "EndTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
               "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
            }
         }
      },
      "traits": {
         "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
      }
   },
   "com.amazonaws.licensemanager#LicenseConversionTaskId": {
      "type": "string",
      "traits": {

       "smithy.api#length": {
            "min": 0,
            "max": 50
         },
         "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
      }
   },
   "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
      "type": "enum",
      "members": {
         "IN_PROGRESS": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "IN_PROGRESS"
            }
         },
         "SUCCEEDED": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "SUCCEEDED"
            }
```

```
          },
          "FAILED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTasks":
{
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTask"
        }
      },
      "com.amazonaws.licensemanager#LicenseCountingType": {
        "type": "enum",
        "members": {
          "VCPU": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "vCPU"
            }
          },
          "INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Instance"
            }
          },
          "CORE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Core"
            }
          },
          "SOCKET": {
            "target": "smithy.api#Unit",
             "traits": {
              "smithy.api#enumValue": "Socket"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseDeletionStatus": {
        "type": "enum",
        "members": {
```

```
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    }
  },
  "com.amazonaws.licensemanager#LicenseList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#License"
    }
  },
  "com.amazonaws.licensemanager#LicenseOperationFailure": {

    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Resource type.</p>"
        }
      },
      "ErrorMessage": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Error message.</p>"
        }
      },
      "FailureTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {

"smithy.api#documentation": "<p>Failure time.</p>"
        }
```

```
        },
        "OperationName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the operation.</p>"
          }
        },
        "ResourceOwnerId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
          }
        },
        "OperationRequestedBy": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
          }
        },
        "MetadataList": {

          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Reserved.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailureList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
      }
    },
    "com.amazonaws.licensemanager#LicenseSpecification": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "AmiAssociationScope": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseSpecifications": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseSpecification"
      }
    },
    "com.amazonaws.licensemanager#LicenseStatus": {
      "type": "enum",
      "members": {
        "AVAILABLE": {
          "target": "smithy.api#Unit",

          "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "PENDING_AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_AVAILABLE"
          }
        },
        "DEACTIVATED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DEACTIVATED"
          }
        },
        "SUSPENDED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUSPENDED"
          }
```

```
      },
      "EXPIRED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "EXPIRED"
        }
      },

     "PENDING_DELETE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_DELETE"
        }
      },
      "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DELETED"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#LicenseUsage": {
    "type": "structure",
    "members": {
      "EntitlementUsages": {
        "target": "com.amazonaws.licensemanager#EntitlementUsageList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlement usages.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

    }
  },
  "com.amazonaws.licensemanager#LicenseUsageException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "LicenseUsageFailure",
        "httpResponseCode": 412
```

          },
          "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 412
        }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
      },
      "output": {

    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n   <p>Resource associations need not consume licenses from a license
configuration. \n   For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",

```
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
        "type": "structure",
        "members": {
          "LicenseConfigurationAssociations": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
            "traits": {
              "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
            }
          },
          "NextToken": {

"target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListDistributedGrants": {
        "type": "operation",
        "input": {
```

```
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
      "type": "structure",
      "members": {
        "GrantArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {

        "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n    <ul>\n      <li>\n        <p>\n          <code>LicenseArn</code>\n        </p>\n
      </li>\n      <li>\n        <p>\n          <code>GrantStatus</code>\n        </p>\n        </li>\n
```

```
            <li>\n          <p>\n              <code>GranteePrincipalARN</code>\n          </p>\n        </li>\n
<li>\n          <p>\n              <code>ProductSKU</code>\n          </p>\n        </li>\n        <li>\n
<p>\n              <code>LicenseIssuerName</code>\n
       </p>\n        </li>\n        </ul>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
        "type": "structure",
        "members": {
            "Grants": {
                "target": "com.amazonaws.licensemanager#GrantList",

    "traits": {
                    "smithy.api#documentation": "<p>Distributed grant details.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
        },
        "errors": [
          {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
```

```
          },
        "traits": {
          "smithy.api#input": { }
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
        "type": "structure",
        "members": {
          "LicenseOperationFailureList": {
            "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
            "traits": {
              "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": { }
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
        },

  "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
```

```
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configurations
 for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n          Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n          Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n
<p>\n            <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n          Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
   </li>\n        <li>\n        <p>\n            <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n          Logical operators are
```

```
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n       </ul>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
        "type": "structure",
        "members": {
            "LicenseConfigurations": {
                "target": "com.amazonaws.licensemanager#LicenseConfigurations",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license configurations.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",

                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
```

```
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
          "type": "structure",
          "members": {
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#Filters",
              "traits": {
                "smithy.api#documentation": "<p>\n      Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n    </p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
          "type": "structure",
          "members": {
            "LicenseConversionTasks": {
              "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
              "traits": {
                "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
              }
            },
```

```
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target":
"com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
```

```
            }
        },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n          <ul>\n            <li>\n              <p>\n                <code>LicenseConfigurationArn</code>\n
</p>\n            </li>\n          </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits":
{
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
        "type": "structure",
        "members": {
          "ReportGenerators": {
            "target": "com.amazonaws.licensemanager#ReportGeneratorList",
            "traits": {
              "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
```

```
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an associated license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
```

```
            "traits": {

      "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {
        "LicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
```

```
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
```

```
            "target": "com.amazonaws.licensemanager#LicenseList",
            "traits": {
              "smithy.api#documentation": "<p>License details.</p>"
            }
          },

"NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenses": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicensesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicensesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
      }
```

        },
        "com.amazonaws.licensemanager#ListLicensesRequest": {
            "type": "structure",
            "members": {
                "LicenseArns": {
                    "target": "com.amazonaws.licensemanager#ArnList",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
                    }
                },

    "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>Beneficiary</code>\n   </p>\n
        \n   \n   <p>\n   <code>ProductSKU</code>\n   </p>\n   \n
    \n   <p>\n   <code>Fingerprint</code>\n   </p>\n   \n   \n
<p>\n   <code>Status</code>\n   </p>\n   \n   "
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token
 for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListLicensesResponse": {
            "type": "structure",
            "members": {
                "Licenses": {
                    "target": "com.amazonaws.licensemanager#LicenseList",
                    "traits": {
                        "smithy.api#documentation": "<p>License details.</p>"
                    }
                },

```
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrants": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n        recipient as this Amazon Web Services account, your organization, or an organizational unit\n
   (OU) to which this member account belongs.</p>"
        }
```

```
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target":
"com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Filters": {
```

```
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
    <ul>\n        <li>\n            <p>\n                <code>ParentArn</code>\n            </p>\n        </li>\n
<li>\n            <p>\n                <code>GranteePrincipalArn</code>\n            </p>\n        </li>\n    </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
            "type": "structure",
            "members": {
                "Grants": {
                    "target": "com.amazonaws.licensemanager#GrantList",
                    "traits": {
                        "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
            "type": "structure",
```

```
        "members": {
          "GrantArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
             "traits": {
               "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
             }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
               "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n      <ul>\n        <li>\n          <p>\n            <code>ProductSKU</code>\n          </p>\n
      </li>\n        <li>\n          <p>\n            <code>LicenseIssuerName</code>\n          </p>\n
</li>\n        <li>\n          <p>\n            <code>LicenseArn</code>\n          </p>\n        </li>\n
<li>\n          <p>\n            <code>GrantStatus</code>\n          </p>\n        </li>\n        <li>\n
<p>\n
      <code>GranterAccountId</code>\n          </p>\n        </li>\n      </ul>"
             }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
               "smithy.api#documentation": "<p>Token for the next set of results.</p>"
             }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
               "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
             }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
        "type": "structure",
        "members": {
          "Grants": {
            "target": "com.amazonaws.licensemanager#GrantList",
             "traits": {
               "smithy.api#documentation": "<p>Received grant details.</p>"
             }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#ListReceivedLicenses": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
       },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists received licenses.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
{
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {

          "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n          <p>\n            <code>Beneficiary</code>\n          </p>\n
  </li>\n        <li>\n          <p>\n            <code>ProductSKU</code>\n
      </p>\n        </li>\n        </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
```

```
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#MaxSize100",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
    "type": "structure",
    "members": {
      "Licenses": {
        "target": "com.amazonaws.licensemanager#GrantedLicenseList",
        "traits": {
          "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
    "type": "structure",
    "members": {
      "LicenseArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
        }
      },

      "Filters": {
        "target": "com.amazonaws.licensemanager#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>ProductSKU</code>\n                </p>\n
```

```
        </li>\n        <li>\n          <p>\n          <code>Status</code>\n        </p>\n      </li>\n
<li>\n      <p>\n          <code>Fingerprint</code>\n        </p>\n      </li>\n      <li>\n
<p>\n          <code>IssuerName</code>\n        </p>\n      </li>\n      <li>\n        <p>\n
<code>Beneficiary</code>\n        </p>\n      </li>\n    </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
            "type": "structure",
            "members": {
                "Licenses": {
                    "target": "com.amazonaws.licensemanager#GrantedLicenseList",
                    "traits": {
                        "smithy.api#documentation": "<p>Received license details.</p>"
                    }
                },

                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListResourceInventory": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FailedDependencyException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#InventoryFilterList",
```

                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n   are supported:</p>\n   \n   \n   <p>\n   <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n   Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
        \n   \n   <p>\n   <code>application_name</code> - The name of the
application. \n   Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
    \n   \n   <p>\n   <code>license_included</code> - The type of license included. \n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n   Possible values
are <code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-
server-web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n
    <p>\n   <code>platform</code> - The platform of the resource.\n   Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n   \n   \n   <p>\n
    <code>resource_id</code> - The ID of the resource. \n   Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   \n   <p>\n
    <code>tag:<key></code> - The key/value combination of a tag assigned \n   to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n   <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n   \n   "
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        }
    },
    "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
        "type": "structure",
        "members": {
            "ResourceInventoryList": {
                "target": "com.amazonaws.licensemanager#ResourceInventoryList",
                "traits": {
                    "smithy.api#documentation": "<p>Information
about the resources.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListTagsForResource": {
        "type": "operation",

```json
      "input": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

    "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
```

```
              "target": "com.amazonaws.licensemanager#TagList",
              "traits": {
                "smithy.api#documentation": "<p>Information about the tags.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListTokens": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListTokensRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListTokensResponse"
        },
        "errors":
[
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists your tokens.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListTokensRequest": {
        "type": "structure",
        "members": {
          "TokenIds": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>Token IDs.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n        <li>\n            <p>\n                <code>LicenseArns</code>\n            </p>\n        </li>\n
</ul>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
            "Tokens": {
                "target": "com.amazonaws.licensemanager#TokenList",
                "traits": {
                    "smithy.api#documentation": "<p>Received token details.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
```

```
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying license\n      consumption details by resource at a selected point in time. Use this action to audit the\n      current license consumption for any license inventory and configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
```

```
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#Filters",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
      Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n
  <li>\n        <p>\n        <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n        <p>\n
    <code>resourceAccount</code> - The ID of the account that owns the resource. \n        Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        </ul>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
},
"com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
  "type": "structure",
  "members": {

"LicenseConfigurationUsageList": {
      "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
      "traits": {
        "smithy.api#documentation": "<p>Information about the license configurations.</p>"
      }
    },
    "NextToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    }
```

```
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Location": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Long": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#ManagedResourceSummary": {
      "type": "structure",
      "members": {
        "ResourceType":
{
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
          }
        },
        "AssociationCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
      }
    },
    "com.amazonaws.licensemanager#MaxSize100": {

      "type": "integer",
      "traits": {
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
```

```
"com.amazonaws.licensemanager#MaxSize3StringList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 3
        }
    }
},
"com.amazonaws.licensemanager#Message": {
    "type": "string"
},
"com.amazonaws.licensemanager#Metadata": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The key name.</p>"
            }
        },

"Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
    }
},
"com.amazonaws.licensemanager#MetadataList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#Metadata"
    }
},
"com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
```

```
            },
          "traits": {
            "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#Options": {
          "type": "structure",
          "members": {
            "ActivationOverrideBehavior": {
              "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
              "traits": {
                "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n        Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n        <ul>\n            <li>\n                <p>As a license administrator, you can
optionally specify an\n                <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
</li>\n            <li>\n                <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n                account in your
organization.</p>\n            </li>\n            <li>\n                <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n                <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n            are activating the grant.</p>\n            </li>\n        </ul>\n        <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n            <dd>\n                <p>Use this value to activate a grant
without replacing any member accounts\n                active grants for the same product.</p>\n            </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n            <dd>\n                <p>Use this value to activate a
grant and disable other active grants in any\n                member accounts
for the same product. This action will also replace their\n                previously activated grants with this activated
grant.</p>\n            </dd>\n        </dl>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n        override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager</a> in the <i>License Manager User Guide</i>.</p>"
          }
        },
        "com.amazonaws.licensemanager#OrganizationConfiguration": {
          "type": "structure",
          "members": {
            "EnableIntegration": {
              "target": "com.amazonaws.licensemanager#Boolean",
              "traits": {
                "smithy.api#default": false,
                "smithy.api#documentation":
```

"<p>Enables Organizations integration.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
    }
},
"com.amazonaws.licensemanager#PrincipalArnList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#Arn"
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 1
        }
    }
},
"com.amazonaws.licensemanager#ProductInformation": {
    "type": "structure",
    "members": {
        "ResourceType": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Resource type. The
possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                "smithy.api#required": {}
            }
        },
        "ProductInformationFilterList": {
            "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
            "traits": {
                "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n   <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n   <p>The following filters and are supported when
the resource type \n   is <code>SSM_MANAGED</code>:</p>\n
        \n   \n   <p>\n   <code>Application Name</code> - The name of the
application.\n   Logical operator is <code>EQUALS</code>.</p>\n   \n   \n
<p>\n   <code>Application Publisher</code> - The publisher of the application.\n   Logical
operator is <code>EQUALS</code>.</p>\n   \n   \n   <p>\n
<code>Application Version</code> - The version of the application.\n   Logical operator is
<code>EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>Platform

Name</code> - The name of the platform.\n   Logical operator is <code>EQUALS</code>.</p>\n
\n   \n   <p>\n   <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
          \n   \n   <p>\n   <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>. The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n   </p>\n   \n   \n   <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n   Logical operator is <code>NOT_EQUALS</code>.</p>\n   \n
\n   <p>\n   <code>License Included</code> - The type of license included.\n   Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n   Possible values are:
 <code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-server-
web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n
<p>The following filters and logical operators are supported when the resource type\n   is
<code>RDS</code>:</p>\n   \n   \n   <p>\n   <code>Engine Edition</code> -
The edition of the database engine.\n   Logical operator is <code>EQUALS</code>.\n   Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n   \n   \n   <p>\n   <code>License Pack</code> - The license
pack.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>data
guard</code> | \n   <code>diagnostic pack sqlt</code> | \n
          <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n   <code>olap</code>.</p>\n
\n   ",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#ProductInformationFilter": {
          "type": "structure",
          "members": {
            "ProductInformationFilterName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Filter name.</p>",
                "smithy.api#required": {}
              }
            },
            "ProductInformationFilterValue": {
              "target": "com.amazonaws.licensemanager#StringList",

              "traits": {
                "smithy.api#documentation": "<p>Filter value.</p>"
              }
            },

```
      "ProductInformationFilterComparator": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Logical operator.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes product information filters.</p>"
    }
  },
  "com.amazonaws.licensemanager#ProductInformationFilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ProductInformationFilter"
    }
  },
  "com.amazonaws.licensemanager#ProductInformationList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ProductInformation"
    }
  },
  "com.amazonaws.licensemanager#ProvisionalConfiguration": {
    "type": "structure",
    "members": {
      "MaxTimeToLiveInMinutes": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#RateLimitExceededException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
```

```
          "aws.protocols#awsQueryError": {
            "code": "RateLimitExceeded",
            "httpResponseCode": 429
          },
          "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 429
        }
    },
    "com.amazonaws.licensemanager#ReceivedMetadata": {
      "type": "structure",
      "members": {
        "ReceivedStatus": {
          "target": "com.amazonaws.licensemanager#ReceivedStatus",
          "traits": {
            "smithy.api#documentation": "<p>Received status.</p>"
          }
        },
        "ReceivedStatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Received
status reason.</p>"
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReceivedStatus": {
      "type": "enum",
      "members": {
        "PENDING_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_WORKFLOW"
          }
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
```

```
        "traits": {
          "smithy.api#enumValue":
"PENDING_ACCEPT"
        }
      },
      "REJECTED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "REJECTED"
        }
      },
      "ACTIVE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "ACTIVE"
        }
      },
      "FAILED_WORKFLOW": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "FAILED_WORKFLOW"
        }
      },
      "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DELETED"
        }
      },
      "DISABLED": {
        "target": "smithy.api#Unit",
         "traits": {
          "smithy.api#enumValue": "DISABLED"
        }
      },
      "WORKFLOW_COMPLETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "WORKFLOW_COMPLETED"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#RedirectException": {
    "type": "structure",
    "members": {
      "Location": {
        "target": "com.amazonaws.licensemanager#Location",
```

```
          "traits": {
              "smithy.api#httpHeader": "Location"
          }
        },
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
         "smithy.api#error": "client",
         "smithy.api#httpError": 308
      }
    },
    "com.amazonaws.licensemanager#RejectGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#RejectGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#RejectGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#RejectGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#RejectGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#RenewType":
{
        "type": "enum",
        "members": {
```

```
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "WEEKLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Weekly"
          }
        },
        "MONTHLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Monthly"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportContext": {
      "type": "structure",
      "members": {
        "licenseConfigurationArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {

      "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReportFrequency": {
      "type": "structure",
      "members": {
        "value": {
          "target": "com.amazonaws.licensemanager#Integer",
          "traits": {
            "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
          }
        },
        "period": {
```

```
                    "target": "com.amazonaws.licensemanager#ReportFrequencyType",

                    "traits": {
                        "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
            }
        },
        "com.amazonaws.licensemanager#ReportFrequencyType": {
            "type": "enum",
            "members": {
                "DAY": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "DAY"
                    }
                },
                "WEEK": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "WEEK"
                    }
                },
                "MONTH": {
                    "target": "smithy.api#Unit",
                    "traits":
{
                        "smithy.api#enumValue": "MONTH"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#ReportGenerator": {
            "type": "structure",
            "members": {
                "ReportGeneratorName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Name of the report generator.</p>"
                    }
                },
                "ReportType": {
                    "target": "com.amazonaws.licensemanager#ReportTypeList",
                    "traits": {
```

```
            "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
          }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
          }
        },
        "ReportFrequency": {
          "target": "com.amazonaws.licensemanager#ReportFrequency",
          "traits": {
            "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
          }
        },
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
          }
        },
        "LastRunStatus": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
          }
        },
        "LastRunFailureReason":
{
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
          }
        },
        "LastReportGenerationTime": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
          }
        },
        "ReportCreatorAccount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
          }
        },
```

```
        "Description": {
          "target": "com.amazonaws.licensemanager#String",

    "traits": {
            "smithy.api#documentation": "<p>Description of the report generator.</p>"
          }
        },
        "S3Location": {
          "target": "com.amazonaws.licensemanager#S3Location",
          "traits": {
            "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
          }
        },
        "CreateTime": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the report was created.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
          }

    }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportGenerator"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorName": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "com.amazonaws.licensemanager#ReportType": {
```

```
      "type": "enum",
      "members": {
        "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
          }

        },
        "LICENSE_CONFIGURATION_USAGE_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationUsageReport"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportTypeList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportType"
      }
    },
    "com.amazonaws.licensemanager#ResourceInventory": {
      "type": "structure",
      "members": {
        "ResourceId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the resource.</p>"
          }
        },
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits":

{
            "smithy.api#documentation": "<p>Type of resource.</p>"
          }
        },
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },
        "Platform": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
              "smithy.api#documentation": "<p>Platform of the resource.</p>"
            }
          },
          "PlatformVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
            }
          },
          "ResourceOwningAccountId":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about a resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#ResourceInventoryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#ResourceInventory"
        }
      },
      "com.amazonaws.licensemanager#ResourceLimitExceededException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "ResourceLimitExceeded",

            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#ResourceNotFoundException": {
        "type": "structure",
        "members": {
```

```
    "Message": {
      "target": "com.amazonaws.licensemanager#Message"
    }
  },
  "traits": {
    "aws.protocols#awsQueryError": {
      "code": "InvalidResource.NotFound",
      "httpResponseCode": 400
    },
    "smithy.api#documentation": "<p>The resource cannot be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 400
  }
},
"com.amazonaws.licensemanager#ResourceType": {
  "type": "enum",
  "members": {
    "EC2_INSTANCE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EC2_INSTANCE"
      }
    },
    "EC2_HOST": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EC2_HOST"
      }
    },
    "EC2_AMI": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EC2_AMI"
      }
    },
    "RDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "RDS"
      }
    },
    "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
      }
    }
  }
```

```
        },
        "com.amazonaws.licensemanager#S3Location": {
          "type": "structure",
          "members": {
            "bucket": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
              }
            },
            "keyPrefix": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
          }
        },
        "com.amazonaws.licensemanager#ServerInternalException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "InternalError",
              "httpResponseCode": 500
            },
            "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
          }
        },
        "com.amazonaws.licensemanager#SignedToken": {
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 4096
            }
          }
        },
        "com.amazonaws.licensemanager#StatusReasonMessage": {
```

```
        "type": "string",
        "traits":
{
            "smithy.api#length": {
               "min": 0,
               "max": 400
            },
            "smithy.api#pattern": "^[\\s\\S]+$"
         }
      },
      "com.amazonaws.licensemanager#String": {
         "type": "string"
      },
      "com.amazonaws.licensemanager#StringList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanager#String"
         }
      },
      "com.amazonaws.licensemanager#Tag": {
         "type": "structure",
         "members": {
            "Key": {
               "target": "com.amazonaws.licensemanager#String",
               "traits": {
                  "smithy.api#documentation": "<p>Tag key.</p>"
               }
            },
            "Value": {
               "target": "com.amazonaws.licensemanager#String",
               "traits": {
                  "smithy.api#documentation":
"<p>Tag value.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
         }
      },
      "com.amazonaws.licensemanager#TagKeyList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanager#String"
         }
      },
      "com.amazonaws.licensemanager#TagList": {
         "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanager#Tag"
        }
    },
    "com.amazonaws.licensemanager#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#TagResourceResponse"
      },
      "errors": [
        {
          "target":
"com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#TagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits":
{
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
```

```
          "traits": {
            "smithy.api#documentation": "<p>One or more tags.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#TokenData": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
        "TokenType": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "ExpirationTime": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
format.</p>"
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
```

```
        "traits": {
          "smithy.api#documentation": "<p>Data specified by the caller.</p>"
        }
      },
      "RoleArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
        }
      }
    },
    "traits": {

      "smithy.api#documentation": "<p>Describes a token.</p>"
    }
  },
  "com.amazonaws.licensemanager#TokenList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#TokenData"
    }
  },
  "com.amazonaws.licensemanager#TokenString": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 4096
      },
      "smithy.api#pattern": "^\\S+$"
    }
  },
  "com.amazonaws.licensemanager#TokenType": {
    "type": "enum",
    "members": {
      "REFRESH_TOKEN": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "REFRESH_TOKEN"
        }
```

```
          }
        }
      },
      "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
 {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
        }
      },
```

```
        "com.amazonaws.licensemanager#UntagResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanager#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
          },
          "output": {

"target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseConfigurationStatus": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
          "traits": {
            "smithy.api#documentation": "<p>New status of the license configuration.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {

    "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
    configuration is licenseAffinityToHost.</p>"
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
```

```
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                   "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
                }
             },
             "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                   "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
                }
             },
             "Description": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                   "smithy.api#documentation": "<p>New description of the license configuration.</p>"
                }
             },
             "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                   "smithy.api#documentation": "<p>New product information.</p>"
                }
             },
             "DisassociateWhenNotFound": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                   "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
             }
          },
          "traits":
{
             "smithy.api#input": {}
          }
       },
       "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
          "type": "structure",
          "members": {},
          "traits": {
             "smithy.api#output": {}
          }
       },
       "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
          "type": "operation",
          "input": {
             "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportGeneratorName": {
```

```
              "target": "com.amazonaws.licensemanager#ReportGeneratorName",
              "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>",
                "smithy.api#required": {}
              }
            },
            "Type": {
              "target": "com.amazonaws.licensemanager#ReportTypeList",
              "traits": {
                "smithy.api#documentation":
"<p>Type of reports to generate. The following report types are supported:</p>\n          <ul>\n            <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n            </li>\n            <li>\n            <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n            </li>\n          </ul>",
                "smithy.api#required": {}
              }
            },
            "ReportContext": {
              "target": "com.amazonaws.licensemanager#ReportContext",
              "traits": {
                "smithy.api#documentation": "<p>The report context.</p>",
                "smithy.api#required": {}
              }
            },
            "ReportFrequency": {
              "target": "com.amazonaws.licensemanager#ReportFrequency",
              "traits":
{
                "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                "smithy.api#required": {}
              }
            },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientRequestToken",
              "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
              }
            }
          },
          "traits": {
```

```
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },

            {
                "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
            },
            {
                "target": "com.amazonaws.licensemanager#LicenseUsageException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
```

```
          }
        },

    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
              "smithy.api#required": {}
            }
          },
          "AddLicenseSpecifications": {
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
              "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
            }
          },
          "RemoveLicenseSpecifications": {
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
              "smithy.api#documentation":
"<p>ARNs of the license configurations to remove.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
    "com.amazonaws.licensemanager#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
        },
        "errors": [
```

```
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
    }
  },
  "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "S3BucketArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
        }
      },
      "SnsTopicArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
        }
      },
      "OrganizationConfiguration": {
        "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
        }
      },
      "EnableCrossAccountsDiscovery": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
```

```
                    "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

        }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UsageOperation": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            }
        }
    },
    "com.amazonaws.licensemanager#ValidationException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

     "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    }
  }
}
```

```
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager


import (
 "context"
 "fmt"
```

```go
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
	if params == nil {
		params = &DeleteLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*DeleteLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type DeleteLicenseInput struct {

	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// Current version
	of the license.
	//
	// This member is required.
	SourceVersion *string

	noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

	// Date when the license is deleted.
	DeletionDate *string

	// License status.
	Status types.LicenseDeletionStatus
```

```go
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack,
options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
        return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
        return err
    }
```

```go
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "DeleteLicense",
	}
}
```

1.28 aws-aws-sdk-go-v2-service-ssooidc 1.21.1

1.28.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil,
 err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
```

```go
	// This member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to associate the license with.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

	// A structure containing data about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
	if err !=
		nil {
		return err
	}
	err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
```

```
   return err
   }
   if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
    return err
   }
   if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
   }
   if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
   }
   if err = addRetryMiddlewares(stack, options); err != nil {
    return err
   }
   if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
   }
   if err = awsmiddleware.AddRecordResponseTiming(stack);
   err != nil {
    return err
   }
   if err = addClientUserAgent(stack, options); err != nil {
    return err
   }
   if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
   }
   if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
   }
   if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
   }
   if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
   }
   if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
   middleware.Before); err != nil {
    return err
   }
   if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
   }
   if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
   }
   if err = addResponseErrorMiddleware(stack); err != nil {
    return err
```

```
      }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack,
    options); err != nil {
     return err
    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

name: License Scan

on: [pull_request]

```
jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
    matrix:
      python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
        path: sdkbase
        ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
        path: new-ref
        fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
```

```yaml
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${{ matrix.python-version }}
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
      # compare
      - name: License test
        run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)


// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options)) (*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }


 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns, c.addOperationCheckInLicenseMiddlewares)
```

```go
	if err != nil {
	 return nil, err
	}

	out := result.(*CheckInLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type CheckInLicenseInput struct {

	// License consumption token.
	//
	// This member is required.
	LicenseConsumptionToken *string

	// License beneficiary.
	Beneficiary
	*string

	noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
	 return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
	if err != nil {
	 return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
	if err != nil {
	 return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
	 return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
	 return err
```

```
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
    err != nil {
    return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
    return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return
    err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
    middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
```

```
        }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "CheckInLicense",
 }
}
```

```
{
   "smithy": "2.0",
   "shapes": {
     "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
       "type": "integer"
     },
     "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
       "type": "long"
     },
     "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
       "type": "structure",
       "members": {
         "Name": {
           "target": "smithy.api#String",
           "traits": {
             "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
         },
         "Values": {
           "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
           "traits": {
             "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
           }
         },
```

```
      "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
         "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
         }
       }
     },
     "traits": {
       "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
     }
   },
   "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
     "type": "list",
     "member": {
       "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
     }
   },
   "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
     "type": "operation",
     "input": {
       "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
     },
     "output": {

       "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
     },
     "errors": [
       {
         "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
       },
       {
         "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
       },
       {
         "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
       }
     ],
     "traits": {
       "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
       "smithy.api#http": {
         "code": 200,
         "method": "POST",
         "uri": "/subscription/GetServiceSettings"
       },
       "smithy.api#idempotent": {}
     }
   },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
             "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
             }
          },
          "LinuxSubscriptionsDiscoverySettings": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
             "traits": {
                "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n       Organizations integration has been enabled, and
 which Regions data will be aggregated from.</p>"
             }
          },
          "Status": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
             "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
             }
          },
          "StatusMessage": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
             "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
             }
          },
          "HomeRegions": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays
 the aggregated data for Linux\n       subscriptions.</p>"
             }
          }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
```

```
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }

    },
      "AccountID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the instance.</p>"
        }
      },
      "Region": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
        }
      },
      "UsageOperation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n          <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n          operation values</a> in the <i>License Manager User
Guide</i>.</p>"
        }
      },
      "ProductCode": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
                }
            },
            "LastUpdatedTime": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
                }
            },
            "SubscriptionName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
        ],

      "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
            "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id",
              "content-length",
              "x-api-key",
              "authorization",
              "x-amz-date",

  "x-amz-security-token",
              "Access-Control-Allow-Headers",
              "Access-Control-Allow-Methods",
              "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
```

Linux subscriptions on running\n   Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The
  AWS region used to dispatch the request.",
                  "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
  support dual-stack, dispatching the request MAY return an error.",
                  "type": "Boolean"
              },
              "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
  configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

                  "type": "Boolean"
                },
                "Endpoint": {
                  "builtIn": "SDK::Endpoint",
                  "required": false,
                  "documentation": "Override the endpoint used to send this request",
                  "type": "String"
                }
              },
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "isSet",
                      "argv": [
                        {
                          "ref": "Endpoint"
                        }
                      ]
                    }
                  ],

```
"type": "tree",
"rules": [
  {

  "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ],
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
      "type": "error"
  },
  {
      "conditions": [],
      "type": "tree",
      "rules": [
          {

      "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ],
          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
          "type": "error"
        },
        {
          "conditions": [],

"endpoint": {
              "url": {
                "ref": "Endpoint"
              },
              "properties": {},
              "headers": {}
            },
```

```
                                "type": "endpoint"
                            }
                        ]
                    }
                ]
            },
            {
                "conditions": [],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "isSet",

                                "argv": [
                                    {
                                        "ref": "Region"
                                    }
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "aws.partition",
                                        "argv": [
                                            {
                                                "ref": "Region"
                                            }
                                        ],

                                        "assign": "PartitionResult"
                                    }
                                ],
                                "type": "tree",
                                "rules": [
                                    {
                                        "conditions": [
                                            {
                                                "fn": "booleanEquals",
                                                "argv": [
                                                    {
                                                        "ref": "UseFIPS"
                                                    },
                                                    true
```

```
                          ]
                        },

            {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [
                      {

                  "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]

      }
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },

                  "supportsDualStack"
                            ]
                          }
```

```
                                    ]
                                  }
                                ],
                                "type": "tree",
                                "rules": [
                                  {
                                    "conditions": [],
                                    "type": "tree",
                                    "rules": [
                                      {

                  "conditions": [],
                                          "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                          },
                                          "type": "endpoint"
                                      }
                                    ]
                                  }
                                ]

                    },
                              {
                                "conditions": [],
                                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                                "type": "error"
                              }
                            ]
                          },
                          {
                            "conditions": [
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  {

                    "ref": "UseFIPS"
                                  },
                                  true
                                ]
                              }
                            ],
                            "type": "tree",
                            "rules": [
```

```
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,

{

                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                }
            ]
        }
    ],
    "type": "tree",

    "rules": [
        {
            "conditions": [],
            "type": "tree",
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                        "properties": {},
                        "headers": {}

                    },
                    "type": "endpoint"
                }
            ]
        },
        {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"
        }
    ]
```

```
        },
                                {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
                                        },
                                        true
                                    ]
                                }
                            ],
                            "type": "tree",
                            "rules": [
                                {

                            "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
                                            true,
                                            {
                                                "fn": "getAttr",
                                                "argv": [
                                                    {
                                                        "ref": "PartitionResult"
                                                    },
                                                    "supportsDualStack"
                                                ]
                                            }
                                        ]
                                    }
                                ],
                                "type": "tree",
                                "rules": [
                                    {
                                        "conditions": [],
                                        "type": "tree",
                                        "rules": [
                                            {
                                                "conditions": [],

                                    "endpoint": {
                                                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                "properties": {},
                                                "headers": {}
```

                                            },
                                          "type": "endpoint"
                                        }
                                      ]
                                    }
                                  ]
                                },

            {
                                          "conditions": [],
                                          "error": "DualStack is enabled but this partition does not support DualStack",
                                          "type": "error"
                                        }
                                      ]
                                    },
                                    {
                                      "conditions": [],
                                      "type": "tree",
                                      "rules": [
                                        {
                                          "conditions": [],
                                          "endpoint": {
                                            "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                          },
                                          "type": "endpoint"
                                        }
                                      ]
                                    }
                                  ]
                                }
                              ]
                            },
                            {
                              "conditions": [],
                              "error": "Invalid Configuration: Missing Region",
                              "type": "error"
                            }
                          ]
                        }
                      ]
                    },
                    "smithy.rules#endpointTests": {
                      "testCases": [
                        {
                            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

```
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {

"endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
```

```
          "UseDualStack": false
        }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {


"url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
         "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint":
```

```
{
                    "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-gov-east-1",
                  "UseFIPS": true,
                   "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-gov-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint":
{

                    "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-gov-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
```

```
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": true
```

```
                }
            },
            {

    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
                    }
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {

              "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled but this partition does not support DualStack"
                },
                "params": {
```

```
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
               }
            },
            {
               "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
               "expect": {
                  "endpoint": {
                     "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
               },
               "params": {
                  "Region": "us-isob-east-1",

"UseFIPS": false,
                  "UseDualStack": false
               }
            },
            {
               "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
               "expect": {
                  "endpoint": {
                     "url": "https://example.com"
                  }
               },
               "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
               }
            },
            {
               "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",

"expect": {
                  "endpoint": {
                     "url": "https://example.com"
                  }
               },
               "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
               }
            },
```

```json
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,

    "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "Missing region",
            "expect": {
              "error": "Invalid Configuration: Missing Region"
            }
          }

      }
        ],
        "version": "1.0"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled LinuxSubscriptionsDiscovery"
        },
        {
          "name": "Disabled",
```

```
                    "value": "Disabled",
                    "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
        "type": "structure",
        "members": {
            "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "OrganizationIntegration": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
                "traits": {
                    "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type":
"operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
                "smithy.api#http": {

                    "code": 200,
                    "method": "POST",
                    "uri": "/subscription/ListLinuxSubscriptionInstances"
                },
                "smithy.api#idempotent": {},
                "smithy.api#paginated": {
                    "inputToken": "NextToken",
                    "outputToken": "NextToken",
                    "pageSize": "MaxResults",
                    "items": "Instances"
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
            "type": "structure",
            "members": {
                "Filters": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by
the name of\n    <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n    <p>The valid names for this filter
are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>AmiID</code>\n        </p>\n
</li>\n      <li>\n        <p>\n          <code>InstanceID</code>\n        </p>\n      </li>\n
<li>\n        <p>\n          <code>AccountID</code>\n        </p>\n      </li>\n      <li>\n
<p>\n          <code>Status</code>\n        </p>\n      </li>\n      <li>\n        <p>\n
<code>Region</code>\n        </p>\n      </li>\n      <li>\n        <p>\n
<code>UsageOperation</code>\n        </p>\n      </li>\n      <li>\n
  <p>\n          <code>ProductCode</code>\n        </p>\n      </li>\n      <li>\n        <p>\n
  <code>InstanceType</code>\n        </p>\n      </li>\n    </ul>\n    <p>The valid Operators for this
filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n
</li>\n      <li>\n        <p>\n          <code>equals</code>\n        </p>\n      </li>\n      <li>\n
  <p>\n          <code>Notequal</code>\n        </p>\n      </li>\n    </ul>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
```

```
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
```

```
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n      organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }

      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't
match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n
<li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n
<p>The valid Operators for this filter are:</p>\n
     <ul>\n        <li>\n          <p>\n            <code>contains</code>\n          </p>\n        </li>\n
<li>\n          <p>\n            <code>equals</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
       <code>Notequal</code>\n          </p>\n        </li>\n      </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
```

```
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>",
            "smithy.api#length": {


              "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "EQUAL",
              "value": "Equal",
              "documentation": "Equal operator"
            },
            {
              "name": "NOT_EQUAL",
              "value": "NotEqual",
```

```
              "documentation": "Not equal operator"
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
          "smithy.api#length": {
            "min": 1,
            "max": 20
          }
        }
      },

    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled OrganizationIntegration"
            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled OrganizationIntegration"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [

{
              "name": "InProgress",
              "value": "InProgress",
              "documentation": "InProgress status"
            },
```

```
                    {
                        "name": "Completed",
                        "value": "Completed",
                        "documentation": "Completed status"
                    },
                    {
                        "name": "Successful",
                        "value": "Successful",
                        "documentation": "Successful status"
                    },
                    {
                        "name": "Failed",
                        "value": "Failed",
                        "documentation": "Failed status"
                    }
                ]
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
            "type": "list",
            "member": {
                "target": "smithy.api#String",
                "traits":
{
                    "smithy.api#length": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                },
                "smithy.api#uniqueItems": {}
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
            "type": "map",
            "key": {
                "target": "smithy.api#String"
            },
            "value": {
                "target": "smithy.api#String"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
```

```
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription.</p>"

          }
        },
          "Type": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
            }
          },
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },

    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required": {}
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>",
```

```
          "smithy.api#required": {}
                    }
              },
            "AllowUpdate": {
                "target": "smithy.api#Boolean",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },

            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
          }
        }
      }
    }
}
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
```

```go
  out.ResultMetadata = metadata
  return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //   - Beneficiary
 //   - ProductSKU
 //   - Fingerprint
 //
   - Status
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
```

```
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if
err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
```

```go
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
   return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
   return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
   return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
   return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
   return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
   return err
  }
  return nil
 }

 func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
  return &awsmiddleware.RegisterServiceMetadata{
   Region:        region,
   ServiceID:
     ServiceID,
   OperationName: "ListLicenses",
  }
 }
 // Code generated by smithy-go-codegen DO NOT EDIT.

 package grafana

 import (
  "context"
  "fmt"
  awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
  "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
  "github.com/aws/aws-sdk-go-v2/service/grafana/types"
  "github.com/aws/smithy-go/middleware"
  smithyhttp "github.com/aws/smithy-go/transport/http"
 )

 // Removes the Grafana Enterprise license from a workspace.
 func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
 ...func(*Options)) (*DisassociateLicenseOutput, error) {
```

```go
	if params == nil {
		params = &DisassociateLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*DisassociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type DisassociateLicenseInput struct {

	// The type of license to remove from the workspace.
	//
	//
	This member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to remove the Grafana Enterprise license from.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

	// A structure containing information about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
```

```go
	}
	err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack,
options, "DisassociateLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
		return err
	}
```

```go
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
	return err
	}
	if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
	return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
	return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
	return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
	return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
	return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
	return err
	}
	return nil
	}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
	{
	return &awsmiddleware.RegisterServiceMetadata{
	Region:        region,
	ServiceID:     ServiceID,
	OperationName: "DisassociateLicense",
	}
	}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type
 CheckoutLicenseInput struct {

 // Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
```

```
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource
    Name (ARN) of the checkout license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}
```

```go
func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil
 {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
```

```go
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack);
	err != nil {
		return err
	}
	if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
	middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
```

```
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier
 that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see RFC 7518 Digital Signature with
 // RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData
```

```go
    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}
```

```go
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return
```

```
      err
      }
      if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
      return err
      }
      if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
      return err
      }
      if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
      return err
      }
      if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
      return err
      }
      if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
      middleware.Before); err != nil {
      return err
      }
      if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
      return err
      }
      if err = addRequestIDRetrieverMiddleware(stack); err != nil {
      return err
      }
      if err = addResponseErrorMiddleware(stack); err != nil {
      return err
      }
      if err = addRequestResponseLogging(stack, options); err != nil {
      return err
      }
      if err = addDisableHTTPSMiddleware(stack, options); err != nil {
      return err
      }
      return nil
      }

      func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
      string) *awsmiddleware.RegisterServiceMetadata {
      return &awsmiddleware.RegisterServiceMetadata{
       Region:      region,
       ServiceID:    ServiceID,
       OperationName: "CheckoutBorrowLicense",
      }
      }
      // Code generated by smithy-go-codegen DO NOT EDIT.

      package licensemanager
```

```go
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide
 to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration
```

```go
	// License entitlements.
	//
	// This member is required.
	Entitlements []types.Entitlement

	// Home Region for the license.
	//
	// This member is required.
	HomeRegion *string

	// License issuer.
	//
	// This member is required.
	Issuer *types.Issuer

	// License name.
	//
	// This member is required.
	LicenseName *string

	// Product name.
	//
	// This member is required.
	ProductName *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// Date and time range during which the license is valid, in ISO8601-UTC format.
	//
	// This member is required.
	Validity
	*types.DatetimeRange

	// Information about the license.
	LicenseMetadata []types.Metadata

	noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string
```

```go
	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
		return fmt.Errorf("add
protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
```

```
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
    err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
```

```go
  Region:        region,
  ServiceID:      ServiceID,
  OperationName:
 "CreateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version
```

```go
    *string

    noSmithyDocumentSerde
}

type GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return
    err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
```

```
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack);
err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
return nil
```

```
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "GetLicense",
 }
}
```

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
```

```go
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
  // - ProductSKU
 //  - Status
 //  - Fingerprint
 //  - IssuerName
 //  - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense
```

```go
    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
        return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
        return err
```

```go
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack);
	err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
	middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
	string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "ListReceivedLicenses",
```

}
}

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
 incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
```

"<p>Details about an Active Directory identity provider.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n   <note>\n   <p>Your estimated bill for charges on the number of users and related
costs will take 48\n   hours to appear for billing periods that haven't closed (marked as Pending billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p>\n   </note>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/user/AssociateUser"
                },
                "smithy.api#idempotent": {}
            }

```
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
            "type": "structure",
            "members": {
                "Username": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                        "smithy.api#required": {}
                    }
                },
                "InstanceId": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
                        "smithy.api#required": {}
                    }
                },
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#documentation": "<p>The identity provider of the user.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Domain": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The domain name of the user.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
            "type": "structure",
            "members": {
                "InstanceUserSummary": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                    "traits": {
                        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                        "smithy.api#required": {}
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
            "type": "integer"
```

        },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n   resource.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based

```
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/DeregisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },

    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
```

```
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter name and value pair that is used
to return more specific results from a describe\n    operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n    attributes, or IDs.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
```

```
              "traits": {
                  "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n      settings such as the subnets to provision VPC endpoints.</p>",
                  "smithy.api#required":
  {}
              }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an identity provider.</p>",
                "smithy.api#required": {}
            }
          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          }
      },
      "traits": {
          "smithy.api#documentation":
  "<p>Describes an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
```

```
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status
of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
```

```
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits":
{
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
       "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the user.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",

        "version": "2018-05-10",
        "operations": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
          },
          {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
      }

    },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
       },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
          "x-amzn-trace-id",
          "content-length",
          "x-api-key",
          "authorization",
          "x-amz-date",
          "x-amz-security-token",
          "Access-Control-Allow-Headers",
          "Access-Control-Allow-Methods",
          "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
          "x-amzn-errortype",
```

```
                "x-amzn-requestid",
                "x-amzn-trace-id"


    ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack
 endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }

      },
          "rules": [
            {
              "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
```

```
                  "ref": "Endpoint"
                }
              ]
            }
          ],
          "type": "tree",
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },

                    true
                  ]
                }
              ],
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
              "type": "error"
            },
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
              "type": "error"
            },
            {
              "conditions": [],
              "endpoint": {
                "url": {
                  "ref": "Endpoint"
                },
                "properties": {},
                "headers": {}
              },
```

```
                              "type": "endpoint"
                          }
                      ]
                  },
                  {
                      "conditions": [
                          {
                              "fn": "isSet",
                              "argv": [

          {
                                  "ref": "Region"
                              }
                          ]
                      }
                  ],
                  "type": "tree",
                  "rules": [
                      {
                          "conditions": [
                              {
                                  "fn": "aws.partition",
                                  "argv": [
                                      {
                                          "ref": "Region"
                                      }
                                  ],
                                  "assign": "PartitionResult"
                              }
                          ],
                          "type": "tree",

      "rules": [
                              {
                                  "conditions": [
                                      {
                                          "fn": "booleanEquals",
                                          "argv": [
                                              {
                                                  "ref": "UseFIPS"
                                              },
                                              true
                                          ]
                                      },
                                      {
                                          "fn": "booleanEquals",
                                          "argv": [
                                              {
```

```
                                            "ref": "UseDualStack"

                                },
                                true
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",

            "argv": [

                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsFIPS"
                                            ]
                                        }
                                    ]
                                },
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {

                    "fn": "getAttr",

                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                            ]
                                        }
                                    ]
                                }
                            ],
                            "type": "tree",
                            "rules": [
                                {
```

```
            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ]
                },
                {
                    "conditions": [],
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                        "type": "error"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",

    "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
```

```
                    ]
                  }
                ]
              }
            ],
            "type": "tree",
            "rules": [
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": {},
                  "headers": {}

                },
                "type": "endpoint"
              }
            ]
          },
          {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"
          }
        ]
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  true,
                  {
```

```
                        "fn": "getAttr",
                          "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]
                        }
                      ]
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {

                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "error": "DualStack is enabled but this partition does
 not support DualStack",
                    "type": "error"
                  }
                ]
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
              }
            ]
          }
        ]
```

```
      },
      {
        "conditions":
[],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
      }
    ]
  },
  "smithy.rules#endpointTests": {
    "testCases": [
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
          }
        },
      "params": {
          "Region": "us-east-1",
```

```
              "UseFIPS": false,
              "UseDualStack": true
            }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },

      "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
```

```
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
              }
            },

      "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
```

```
                                "UseFIPS": true,
                                "UseDualStack": false
                            }
                        },
                        {
                            "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
                            "expect": {
                                "endpoint": {
                                    "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                                }
                            },
                            "params": {
                                "Region": "us-gov-east-1",
                                "UseFIPS": false,
                                "UseDualStack": true
                            }
                        },
                        {
                            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                            "expect": {
                                "endpoint": {
                                    "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
                                }
                            },

    "params": {
                                "Region": "us-gov-east-1",
                                "UseFIPS": false,
                                "UseDualStack": false
                            }
                        },
                        {
                            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                            "expect": {
                                "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                            },
                            "params": {
                                "Region": "us-iso-east-1",
                                "UseFIPS": true,
                                "UseDualStack": true
                            }
                        },
                        {
                            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                            "expect":
{
                                "endpoint": {
```

```
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack":
```

true

```
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but
```

this partition does not support one or both"

```
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
```

```
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }

    },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
           "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
```

```
          }
        },
        {

"documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "Missing region",
          "expect": {
            "error": "Invalid Configuration: Missing Region"
          }
        }
      ],
      "version": "1.0"
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated":
{
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "IdentityProviderSummaries"
            }
```

```
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
          "type": "structure",
          "members": {
            "IdentityProviderSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
                "smithy.api#required": {}
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
          },
          "errors": [
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                 }
              },
              "Filters": {
                 "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                 "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
                 }
              }
           }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
           "type":
"structure",
           "members": {
              "InstanceSummaries": {
                 "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                 "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
                 }
              },
              "NextToken": {
                 "target": "smithy.api#String",
                 "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                 }
              }
           }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
           "type": "operation",
           "input": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
           },
           "output": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
           },
           "errors": [
              {
                 "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
              },
              {
                 "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
              },
              {
                 "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
              },
```

```
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListProductSubscriptions"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
```

```
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {

  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n      more sets of key-value pairs that you specify.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
              }
            },

      "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
          },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceUserSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
```

```
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
```

```
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {

        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }

    },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of a product for a user.</p>",
                "smithy.api#required": {}
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
        },
        "SubscriptionStartDate": {
```

```
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
                }
            },
            "SubscriptionEndDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The end date of a subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/RegisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n     subnets to provision VPC endpoints.</p>"
            }
          }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 404
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
      "type": "string",

  "traits": {
        "smithy.api#length": {
          "min": 5,
          "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
```

```
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n      your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n      subnets to provision VPC endpoints, and the security group ID that
 is associated with the VPC\n      endpoints. The security group should permit inbound TCP port 1688
communication from resources\n      in the VPC.</p>"
        }
      },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n        <note>\n          <p>Your estimated bill for charges on the number of users and related costs
will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
 For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n        monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/StartProductSubscription"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
```

```
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
```

```json
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/StopProductSubscription"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required":
{}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation":
  "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {

            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
  request.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
```

```
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity\n    provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
```

```
      "UpdateSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
        "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n      update any combination of settings in a single operation such as the:</p>\n
<ul>\n         <li>\n            <p>Subnets which you want to add to provision VPC endpoints.</p>\n         </li>\n
      <li>\n            <p>Subnets which you want to remove the VPC endpoints from.</p>\n         </li>\n
<li>\n            <p>Security group ID which permits traffic to the VPC endpoints.</p>\n         </li>\n      </ul>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#required": {}
        }
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n      require connectivity to activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID
```

that allows inbound TCP port 1688 communication between resources in\n   your VPC and the VPC endpoints for activation servers.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as\n   the subnets to provision VPC endpoints.</p>"
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
            "type": "structure",
            "members": {
              "message": {
                "target": "smithy.api#String"
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>A parameter is not valid.</p>",
              "smithy.api#error": "client"
            }
          }
        }
      }
    }
    {
      "smithy": "2.0",
      "metadata": {
        "suppressions": [
          {
            "id": "HttpMethodSemantics",
            "namespace": "*"
          },
          {
            "id": "HttpResponseCodeSemantics",
            "namespace": "*"
          },
          {
            "id": "PaginatedTrait",
            "namespace": "*"
          },
          {
            "id": "HttpHeaderTrait",
            "namespace": "*"
          },
          {
            "id": "HttpUriConflict",
            "namespace": "*"
          },

```
        {
            "id": "Service",
            "namespace": "*"
        }
    ]
},
"shapes": {
    "com.amazonaws.licensemanager#AWSLicenseManager": {
        "type": "service",
        "version": "2018-08-01",
        "operations": [
            {
                "target": "com.amazonaws.licensemanager#AcceptGrant"
            },

{
                "target": "com.amazonaws.licensemanager#CheckInLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CheckoutLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateGrant"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateGrantVersion"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
             },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateToken"
            },
```

```
    {
      "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
      "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    }

},
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
    {
      "target": "com.amazonaws.licensemanager#GetGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseUsage"
    },
    {
      "target": "com.amazonaws.licensemanager#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
     },
    {
      "target": "com.amazonaws.licensemanager#ListDistributedGrants"
```

```
    },
    {
      "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
    },
    {

      "target": "com.amazonaws.licensemanager#ListLicenseVersions"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListResourceInventory"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTokens"
    },
    {
      "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
    },
    {
```

```
      "target": "com.amazonaws.licensemanager#RejectGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager",
      "arnNamespace": "license-manager",
      "cloudFormationName": "LicenseManager",
      "cloudTrailEventSource": "licensemanager.amazonaws.com",
      "endpointPrefix": "license-manager"
    },
    "aws.auth#sigv4": {
      "name": "license-manager"
    },
    "aws.protocols#awsJson1_1": {},
    "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n        Amazon Web Services accounts and on-premises servers.</p>",
    "smithy.api#title": "AWS License Manager",
    "smithy.api#xmlNamespace": {
      "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
    },
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {
        "Region": {
          "builtIn": "AWS::Region",
          "required": false,
          "documentation": "The AWS region used to dispatch the request.",
          "type": "String"
        },
```

```
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "Boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
    "type": "Boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "String"
}
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]

      }
    ],
    "type": "tree",
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
```

```
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {

  "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ]
          },
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "type": "tree",
            "rules": [
```

```
{
  "conditions": [
    {
      "fn": "aws.partition",
      "argv": [
        {
          "ref": "Region"
        }
      ],
      "assign": "PartitionResult"
    }
  ],
  "type": "tree",
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        },
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                true,
                {
                  "fn": "getAttr",
                  "argv": [
                    {
```

```
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"type": "tree",
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
]
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or both",
    "type": "error"
}
]
},
```

```json
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        },
```

```
                                    {
                                        "conditions": [],
                                        "error": "FIPS is enabled but this
partition does not support FIPS",
                                        "type": "error"
                                    }
                                ]
                            },
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
                                            {
                                                "ref": "UseDualStack"
                                            },
                                            true
                                        ]
                                    }
                                ],
                                "type": "tree",
                                "rules": [
                                    {
                                        "conditions": [
                                            {
                                                "fn": "booleanEquals",
                                                "argv": [
                                                    true,
                                                    {
                                                        "fn": "getAttr",
                                                        "argv": [
                                                            {
                                                                "ref": "PartitionResult"
                                                            },

    "supportsDualStack"

                                                        ]
                                                    }
                                                ]
                                            }
                                        ],
                                        "type": "tree",
                                        "rules": [
                                            {
                                                "conditions": [],
                                                "endpoint": {
                                                    "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
```

```
                                    "properties": {},
                                    "headers": {}

                        },
                            "type": "endpoint"
                        }
                    ]
                },
                {
                    "conditions": [],
                    "error": "DualStack is enabled but this partition does not support DualStack",
                    "type": "error"
                }
                ]
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
            ]
        }
        ]
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
    {
        "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint":
{
                "url": "https://license-manager.af-south-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "af-south-1",
            "UseFIPS": false,
```

```
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-northeast-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-northeast-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-3.amazonaws.com"
          }
```

```
        },
        "params": {
          "Region": "ap-northeast-3",
          "UseFIPS": false,
          "UseDualStack": false
        }


    },
        {
          "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-1.amazonaws.com"


}
          },
          "params": {
            "Region": "ap-southeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
```

```
                         {

         "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
                   "expect": {
                      "endpoint": {
                         "url": "https://license-manager.ap-southeast-3.amazonaws.com"
                      }
                   },
                   "params": {
                      "Region": "ap-southeast-3",
                      "UseFIPS": false,
                      "UseDualStack": false
                   }
                },
                {
                   "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
                   "expect": {
                      "endpoint": {
                         "url": "https://license-manager.ca-central-1.amazonaws.com"
                      }
                   },

                  "params": {
                      "Region": "ca-central-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                   }
                },
                {
                   "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
                   "expect": {
                      "endpoint": {
                         "url": "https://license-manager.eu-central-1.amazonaws.com"
                      }
                   },
                   "params": {
                      "Region": "eu-central-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                   }
                },
                {
                   "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
                   "expect": {
                      "endpoint": {
                         "url": "https://license-manager.eu-north-1.amazonaws.com"
                      }
```

```
          },
          "params": {
            "Region": "eu-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-west-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
          "expect": {

            "endpoint": {
              "url": "https://license-manager.eu-west-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-west-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
```

```
{
    "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.eu-west-3.amazonaws.com"
        }
    },
    "params": {
        "Region": "eu-west-3",
        "UseFIPS": false,

"UseDualStack": false
        }
    },
    {
        "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager.me-south-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "me-south-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager.sa-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "sa-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager.us-east-1.amazonaws.com"
            }
        },
        "params": {
```

```
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }

    },
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.us-east-2.amazonaws.com"
                }

            },
              "params": {
                "Region": "us-east-2",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.us-east-2.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-east-2",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region
```

us-west-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-west-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-west-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-west-1.amazonaws.com"
              }
            },
            "params": {

        "Region": "us-west-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-west-2.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-west-2",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

    "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-west-2.amazonaws.com"
              }
            },
            "params": {

```
          "Region": "us-west-2",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,

          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-northwest-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
```

```
            "UseDualStack": true
          }
        },

    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-west-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
```

```
                    "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-gov-west-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-fips.us-gov-east-1.api.aws"
                  }
                },

      "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager.us-gov-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
  disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
```

```
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": true,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
                        }
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                        "error": "DualStack is enabled
but this partition does not support DualStack"
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
                        }
                    },
                    "params": {
                        "Region": "us-isob-east-1",
                        "UseFIPS": false,
```

```
                    "UseDualStack": false
                  }
                },

          {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
                  }
                },

      "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "error": "DualStack is enabled but this partition does not support DualStack"
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

  "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
```

```
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,

    "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
```

```
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanager#AcceptGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#AcceptGrantRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
    }
},
"com.amazonaws.licensemanager#AcceptGrantRequest": {
```

```
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#AcceptGrantResponse": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Grant
ARN.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanager#GrantStatus",
              "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#AccessDeniedException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
```

```
        "traits": {
            "aws.protocols#awsQueryError": {

    "code": "ServiceAccessDenied",
                "httpResponseCode": 401
            },
            "smithy.api#documentation": "<p>Access to resource denied.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 401
        }
    },
    "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
        "type": "enum",
        "members": {
            "DISTRIBUTED_GRANTS_ONLY": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
                }
            },
            "ALL_GRANTS_PERMITTED_BY_ISSUER": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#AllowedOperation": {

    "type": "enum",
        "members": {
            "CREATE_GRANT": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CreateGrant"
                }
            },
            "CHECKOUT_LICENSE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CheckoutLicense"
                }
            },
            "CHECKOUT_BORROW_LICENSE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CheckoutBorrowLicense"
```

```
            }
          },
          "CHECK_IN_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CheckInLicense"
            }
          },
          "EXTEND_CONSUMPTION_LICENSE":
{

            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ExtendConsumptionLicense"
            }
          },
          "LIST_PURCHASED_LICENSES": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ListPurchasedLicenses"
            }
          },
          "CREATE_TOKEN": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CreateToken"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#AllowedOperationList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#AllowedOperation"
        },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max":
7
          }
        }
      },
      "com.amazonaws.licensemanager#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 2048
```

```
        },
        "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-
z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
      }
    },
    "com.amazonaws.licensemanager#ArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
      }
    },
    "com.amazonaws.licensemanager#AuthorizationException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError":
{
          "code": "AuthorizationFailure",
          "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n        policy associated with this account.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 403
      }
    },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
      "type": "structure",
      "members": {
        "LastRunTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes automated discovery.</p>"
      }
    },

  "com.amazonaws.licensemanager#Boolean": {
      "type": "boolean",
      "traits": {
```

```json
            "smithy.api#default": false
        }
    },
    "com.amazonaws.licensemanager#BorrowConfiguration": {
        "type": "structure",
        "members": {
            "AllowEarlyCheckIn": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxTimeToLiveInMinutes": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
                    "smithy.api#required": {}
                }
            }

        },
        "traits": {
            "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#BoxBoolean": {
        "type": "boolean"
    },
    "com.amazonaws.licensemanager#BoxInteger": {
        "type": "integer"
    },
    "com.amazonaws.licensemanager#BoxLong": {
        "type": "long"
    },
    "com.amazonaws.licensemanager#CheckInLicense": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanager#AuthorizationException"

        },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
        }
    },
    "com.amazonaws.licensemanager#CheckInLicenseRequest":
 {
        "type": "structure",
        "members": {
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

      },
      "com.amazonaws.licensemanager#CheckInLicenseResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
          },
          {
            "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"

```
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
        "type": "structure",
        "members":
 {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
              "smithy.api#required": {}
            }
          },
          "DigitalSignatureMethod": {
            "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
            "traits": {
              "smithy.api#documentation": "<p>Digital signature method. The possible
 value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
              "smithy.api#required": {}
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          },
          "CheckoutMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about constraints.</p>"
            }
```

```
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },
        "EntitlementsAllowed":
{
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "SignedToken": {
          "target": "com.amazonaws.licensemanager#SignedToken",
          "traits": {
            "smithy.api#documentation": "<p>Signed token.</p>"
          }
        },
```

```
          "IssuedAt": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
            }
          },
          "Expiration": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
            }
          },
          "CheckoutMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about constraints.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
```

```
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Checks out the specified license.</p>\n         <note>\n            <p>If the
account that created the license is the same that is performing the check out, you must\n            specify the account
as the beneficiary.</p>\n         </note>"
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
      "type": "structure",
      "members": {
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product
SKU.</p>",
            "smithy.api#required": {}
          }
        },
        "CheckoutType": {
          "target": "com.amazonaws.licensemanager#CheckoutType",
          "traits": {
            "smithy.api#documentation": "<p>Checkout type.</p>",
            "smithy.api#required": {}
          }
        },
        "KeyFingerprint": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
            "smithy.api#required": {}
          }
        },
```

```
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          }
        },
        "traits": {

    "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
        "type": "structure",
        "members": {
          "CheckoutType": {
            "target": "com.amazonaws.licensemanager#CheckoutType",
            "traits": {
              "smithy.api#documentation": "<p>Checkout type.</p>"
            }
          },
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>"
            }
```

```
        },
        "EntitlementsAllowed": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
          }
        },
        "SignedToken": {
          "target": "com.amazonaws.licensemanager#SignedToken",
          "traits": {
            "smithy.api#documentation": "<p>Signed token.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "IssuedAt": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
          }
        },
        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutType": {
      "type": "enum",
      "members": {
        "PROVISIONAL": {
          "target": "smithy.api#Unit",
```

```json
              "traits": {
                  "smithy.api#enumValue": "PROVISIONAL"
              }
          },
          "PERPETUAL": {
              "target": "smithy.api#Unit",
              "traits": {
                  "smithy.api#enumValue": "PERPETUAL"
              }
          }
      }
  },
  "com.amazonaws.licensemanager#ClientRequestToken": {
      "type": "string",
      "traits": {
          "smithy.api#length": {
              "min": 1,
              "max": 36
          }
      }
  },
  "com.amazonaws.licensemanager#ClientToken": {
      "type": "string",
      "traits": {
          "smithy.api#length": {
              "min": 0,
              "max": 2048
          },
          "smithy.api#pattern": "^\\S+$"
      }
  },
  "com.amazonaws.licensemanager#ConflictException": {
      "type": "structure",
      "members": {
          "Message": {
              "target": "com.amazonaws.licensemanager#Message"
          }
      },
      "traits": {
          "aws.protocols#awsQueryError": {
              "code": "ConflictException",
              "httpResponseCode":
409
          },
          "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 409
```

```
            }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
            "type": "structure",
            "members": {
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#ResourceType",
                    "traits": {
                        "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
                    }
                },
                "ConsumedLicenses": {
                    "target": "com.amazonaws.licensemanager#BoxLong",
                    "traits": {
                        "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
                    }
                }
            },

"traits": {
                "smithy.api#documentation": "<p>Details about license consumption.</p>"
            }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
            }
        },
        "com.amazonaws.licensemanager#ConsumptionConfiguration": {
            "type": "structure",
            "members": {
                "RenewType": {
                    "target": "com.amazonaws.licensemanager#RenewType",
                    "traits": {
                        "smithy.api#documentation": "<p>Renewal frequency.</p>"
                    }
                },
                "ProvisionalConfiguration": {
                    "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
                    }
                },
                "BorrowConfiguration": {
                    "target": "com.amazonaws.licensemanager#BorrowConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
```

```
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#CreateGrant": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CreateGrantRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CreateGrantResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target":
"com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "com.amazonaws.licensemanager#CreateGrantRequest": {
```

```
    "type": "structure",
    "members": {
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientToken",
        "traits": {
          "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant name.</p>",
          "smithy.api#required": {}

  }
      },
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
          "smithy.api#required": {}
        }
      },
      "Principals": {
        "target": "com.amazonaws.licensemanager#PrincipalArnList",
        "traits": {
          "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n        (ARN):</p>\n    <ul>\n        <li>\n            <p>An Amazon Web Services
account, which includes only the account specified.</p>\n        </li>\n    </ul>\n    <ul>\n        <li>\n
    <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n        </li>\n    </ul>\n
<ul>\n
        <li>\n            <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n     </ul>",
          "smithy.api#required": {}
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
        "traits": {
```

```
                    "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
                    "smithy.api#required": {}
                  }
                }
              },
              "traits": {
                "smithy.api#input": {}
              }
            }
          },

          "com.amazonaws.licensemanager#CreateGrantResponse": {
            "type": "structure",
            "members": {
              "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                  "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
              },
              "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                  "smithy.api#documentation": "<p>Grant status.</p>"
                }
              },
              "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Grant version.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#CreateGrantVersion":
          {
              "type": "operation",
              "input": {
                "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
              },
              "output": {
                "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
              },
              "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
       },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n   <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager in the <i>License Manager User Guide</i>.</p>"
    }
  },
  "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
      "type": "structure",
      "members": {
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
          }
        },
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {

```
              "smithy.api#documentation": "<p>Grant name.</p>"
            }
          },
          "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>Grant status reason.</p>"
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the grant.</p>"
            }
          },
          "Options": {
            "target": "com.amazonaws.licensemanager#Options",
            "traits": {
              "smithy.api#documentation": "<p>The options specified for the grant.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    }
  },
  "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Grant ARN.</p>"
        }
      },
      "Status": {
```

```
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
            }
        },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>New version of the grant.</p>"
            }
        }
    },
    "traits":
{
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#CreateLicense": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
```

```
      "traits": {
        "smithy.api#documentation": "<p>Creates a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
 all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of
licenses purchased and used.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>",
```

```
                    "smithy.api#required": {}
                  }
                },
                "Description": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Description of the license configuration.</p>"
                  }
                },
                "LicenseCountingType": {
                  "target": "com.amazonaws.licensemanager#LicenseCountingType",
                  "traits": {
                    "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
                    "smithy.api#required": {}
                  }
                },
                "LicenseCount": {
                  "target": "com.amazonaws.licensemanager#BoxLong",
                  "traits": {
                    "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
                  }
                },
                "LicenseCountHardLimit": {
                  "target": "com.amazonaws.licensemanager#BoxBoolean",
                  "traits": {
                    "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n        blocks the launch of new instances.</p>"
                  }
                },
                "LicenseRules": {
                  "target":
"com.amazonaws.licensemanager#StringList",
                  "traits": {
                    "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n        vary by dimension, as follows.</p>\n
<ul>\n        <li>\n          <p>\n            <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
      <code>licenseAffinityToHost</code> |\n            <code>maximumCores</code> |
<code>minimumCores</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n          <code>maximumCores</code> |
<code>minimumCores</code> |\n            <code>maximumSockets</code> | <code>minimumSockets</code> |\n
      <code>maximumVcpus</code> | <code>minimumVcpus</code>\n          </p>\n        </li>\n
<li>\n
  <p>\n            <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n            <code>maximumSockets</code> |
<code>minimumSockets</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n            <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n          </p>\n        </li>\n        </ul>\n        <p>The unit for
```

<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for <code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n <code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n <code>True</code> and <code>False</code>.</p>"

```
                    }
                },

    "Tags": {
                    "target": "com.amazonaws.licensemanager#TagList",
                    "traits": {
                        "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
                    }
                },
                "DisassociateWhenNotFound": {
                    "target": "com.amazonaws.licensemanager#BoxBoolean",
                    "traits": {
                        "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                    }
                },
                "ProductInformationList": {
                    "target": "com.amazonaws.licensemanager#ProductInformationList",
                    "traits": {
                        "smithy.api#documentation": "<p>Product information.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
            "type": "structure",
            "members": {
                "LicenseConfigurationArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
            "type": "operation",
```

```
"input": {
    "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
},
"output": {
    "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
},
"errors": [
    {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
        "target": "com.amazonaws.licensemanager#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
}
},
"com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {

            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
                "smithy.api#required": {}
            }
        },
        "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
                "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n        For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
```

the <i>License Manager User Guide</i>.</p>",
                  "smithy.api#required": {}
                }
              },
              "DestinationLicenseContext":
{
                  "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                  "traits": {
                      "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI in
the <i>License Manager User Guide</i>.</p>",
                      "smithy.api#required": {}
                  }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
            "type": "structure",
            "members": {
              "LicenseConversionTaskId": {
                  "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

            "traits": {
                    "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {

```
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Creates a report generator.</p>"
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
          "type": "structure",
          "members": {
            "ReportGeneratorName":
{
              "target": "com.amazonaws.licensemanager#ReportGeneratorName",
              "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>",
                "smithy.api#required": {}
              }
            },
            "Type": {
              "target": "com.amazonaws.licensemanager#ReportTypeList",
              "traits": {
                "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n          <ul>\n            <li>\n               <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n            </li>\n            <li>\n               <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n            </li>\n
</ul>",
                "smithy.api#required": {}

              }
            },
```

```
          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits": {
              "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientRequestToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique,
case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn":
{
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
```

```
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseRequest": {
        "type": "structure",
        "members": {
            "LicenseName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License name.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Product
name.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductSKU": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Product SKU.</p>",
                    "smithy.api#required": {}
                }
            },
            "Issuer": {
                "target": "com.amazonaws.licensemanager#Issuer",
                "traits": {
                    "smithy.api#documentation": "<p>License issuer.</p>",
                    "smithy.api#required": {}
                }
            },
            "HomeRegion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Home Region for the license.</p>",
                    "smithy.api#required": {}
```

```
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
          "smithy.api#required": {}
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>",
          "smithy.api#required": {}
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License beneficiary.</p>",
          "smithy.api#required": {}
        }
      },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n          running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n          usage.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
          "smithy.api#documentation": "<p>Information about the license.</p>"
        }
      },
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientToken",
        "traits": {
          "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
          "smithy.api#required": {}
        }
      }
    },
```

```json
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersion": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#ConflictException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target":
"com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>",
            "smithy.api#required": {}
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#Issuer",
```

```
          "traits":
{
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
             "smithy.api#documentation": "<p>Information about the license.</p>"
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",
            "smithy.api#required": {}
          }
        },
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
            "smithy.api#required": {}
          }
        },

  "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the license.</p>"
            }
          }
        },
        "traits": {

"smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New version of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
```

```
{}
          }
        },
      "com.amazonaws.licensemanager#CreateToken": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },

          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateTokenRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
```

```
            "target": "com.amazonaws.licensemanager#Arn",
            "traits":
 {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
                "smithy.api#required": {}
            }
          },
          "RoleArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n        License Manager does not check whether the roles are in use.</p>"
            }
          },
          "ExpirationInDays": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
              "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
            }
          },
          "TokenProperties": {
             "target": "com.amazonaws.licensemanager#MaxSize3StringList",
            "traits": {
              "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n        to the amr claim of the JWT token.</p>"
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateTokenResponse": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
```

```
          "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>"
          }
      },
      "TokenType": {
        "target": "com.amazonaws.licensemanager#TokenType",
        "traits": {
          "smithy.api#documentation": "<p>Token type.</p>"
        }
      },
      "Token": {
        "target": "com.amazonaws.licensemanager#TokenString",
        "traits": {
          "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#DateTime": {
  "type": "timestamp"
},
"com.amazonaws.licensemanager#DatetimeRange": {
  "type": "structure",
  "members": {
    "Begin": {
       "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Start of the time range.</p>",
          "smithy.api#required": {}
        }
    },
    "End": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>End of the time range.</p>"
        }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
  }
},
"com.amazonaws.licensemanager#DeleteGrant": {
  "type": "operation",
  "input": {
```

```
            "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
          }
        },
        "com.amazonaws.licensemanager#DeleteGrantRequest": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
              }
            },
            "StatusReason": {
              "target": "com.amazonaws.licensemanager#StatusReasonMessage",
              "traits": {
                "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
              }
```

```
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the
grant.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
},
"com.amazonaws.licensemanager#DeleteGrantResponse": {
  "type": "structure",
  "members": {
    "GrantArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Grant ARN.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanager#GrantStatus",
      "traits": {
        "smithy.api#documentation": "<p>Grant status.</p>"
      }
    },
    "Version": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation":
"<p>Grant version.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#DeleteLicense": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
```

```
        },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
```

```
                {
                  "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                  "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                  "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
            "type": "structure",
            "members":
 {
                "LicenseConfigurationArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the license configuration.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
            },
            "errors": [
                {
```

              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n   <p>This action
deletes the report generator, which stops it from generating future reports.\n   The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
          "type": "structure",
          "members": {
            "LicenseManagerReportGeneratorArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {

  "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {

```
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the license.</p>",
              "smithy.api#required":
{}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseResponse": {
        "type": "structure",
        "members": {
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
          "DeletionDate": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
```

        },
        "com.amazonaws.licensemanager#DeleteToken": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RedirectException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },

                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
            }
        },
        "com.amazonaws.licensemanager#DeleteTokenRequest": {
            "type": "structure",
            "members": {
                "TokenId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token ID.</p>",
                        "smithy.api#required": {}
                    }
                }

```
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteTokenResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}


            }
        },
        "com.amazonaws.licensemanager#DigitalSignatureMethod": {
            "type": "enum",
            "members": {
                "JWT_PS384": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "JWT_PS384"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#Entitlement": {
            "type": "structure",
            "members": {
                "Name": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Entitlement name.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Value": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
                    }
                },
                "MaxCount": {
                    "target": "com.amazonaws.licensemanager#Long",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
                    }
                },
                "Overage": {
```

```
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement unit.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowCheckIn": {
          "target":
"com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementData": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement data name.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement
data value.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement data unit.</p>",
            "smithy.api#required": {}
          }
        }
```

```
        },
      "traits": {
        "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementDataList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementData"
      }
    },
    "com.amazonaws.licensemanager#EntitlementDataUnit": {
      "type": "enum",
      "members": {
        "COUNT": {
          "target": "smithy.api#Unit",
          "traits": {

"smithy.api#enumValue": "Count"
          }
        },
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "SECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Seconds"
          }
        },
        "MICROSECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Microseconds"
          }
        },
        "MILLISECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Milliseconds"
          }
        },
        "BYTES": {
          "target": "smithy.api#Unit",
```

```
        "traits": {
          "smithy.api#enumValue": "Bytes"
        }
      },
      "KILOBYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes"
        }
      },
      "MEGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes"
        }
      },
      "GIGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes"
        }
      },
      "TERABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabytes"
        }
      },
      "BITS":
{
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits"
        }
      },
      "KILOBITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits"
        }
      },
      "MEGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabits"
        }
      },
      "GIGABITS": {
```

```
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Gigabits"
            }
         },
         "TERABITS": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Terabits"

}
         },
         "PERCENT": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Percent"
            }
         },
         "BYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Bytes/Second"
            }
         },
         "KILOBYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Kilobytes/Second"
            }
         },
         "MEGABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Megabytes/Second"
            }
         },
         "GIGABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
             "traits": {
               "smithy.api#enumValue": "Gigabytes/Second"
            }
         },
         "TERABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "Terabytes/Second"
            }
         },
```

```
      "BITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits/Second"
        }
      },
      "KILOBITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits/Second"
        }
      },
      "MEGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue":
"Megabits/Second"
        }
      },
      "GIGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits/Second"
        }
      },
      "TERABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits/Second"
        }
      },
      "COUNT_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Count/Second"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#EntitlementList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Entitlement"
    }
  },
  "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
    "type": "structure",
```

```
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#EntitlementUnit": {
      "type": "enum",
      "members": {
        "COUNT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count"
          }
        },
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "SECONDS":
{
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Seconds"
          }
        },
        "MICROSECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Microseconds"
          }
        },
        "MILLISECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Milliseconds"
          }
        },
        "BYTES": {
          "target": "smithy.api#Unit",
          "traits": {
```

```
      "smithy.api#enumValue": "Bytes"
    }
  },
  "KILOBYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Kilobytes"
    }
  },
  "MEGABYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Megabytes"
    }
  },
  "GIGABYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Gigabytes"
    }
  },
  "TERABYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Terabytes"
    }
  },
  "BITS": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Bits"
    }
  },
  "KILOBITS": {
    "target": "smithy.api#Unit",
    "traits": {

      "smithy.api#enumValue": "Kilobits"
    }
  },
  "MEGABITS": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Megabits"
    }
  },
  "GIGABITS": {
    "target": "smithy.api#Unit",
```

```
          "traits": {
            "smithy.api#enumValue": "Gigabits"
          }
        },
        "TERABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits"
          }
        },
        "PERCENT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Percent"
          }
        },
        "BYTES_PER_SECOND": {
          "target":
"smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bytes/Second"
          }
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {

"smithy.api#enumValue": "Terabytes/Second"
          }
        },
```

```
    "BITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits/Second"
      }
    },
    "KILOBITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobits/Second"
      }
    },
    "MEGABITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits/Second"
      }
    },
    "GIGABITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabits/Second"
      }
    },

  "TERABITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabits/Second"
      }
    },
    "COUNT_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Count/Second"
      }
    }
  }
},
"com.amazonaws.licensemanager#EntitlementUsage": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Entitlement usage name.</p>",
        "smithy.api#required": {}
      }
```

```
        },
        "ConsumedValue": {
           "target": "com.amazonaws.licensemanager#String",


        "traits": {
             "smithy.api#documentation": "<p>Resource usage consumed.</p>",
             "smithy.api#required": {}
           }
        },
        "MaxCount": {
           "target": "com.amazonaws.licensemanager#String",
           "traits": {
             "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
           }
        },
        "Unit": {
           "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
           "traits": {
             "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
             "smithy.api#required": {}
           }
        }
     },
     "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
     }
  },
  "com.amazonaws.licensemanager#EntitlementUsageList":
{
     "type": "list",
     "member": {
        "target": "com.amazonaws.licensemanager#EntitlementUsage"
     }
  },
  "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
     "type": "operation",
     "input": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
     },
     "output": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
     },
     "errors": [
        {
           "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
           "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>",
            "smithy.api#required": {}

    }
        },
        "DryRun": {
          "target": "com.amazonaws.licensemanager#Boolean",
          "traits": {
            "smithy.api#default": false,
            "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
      "type": "structure",
      "members": {
```

```
      "LicenseConsumptionToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License consumption token.</p>"
        }
      },

      "Expiration": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#FailedDependencyException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      },
      "ErrorCode": {
        "target": "com.amazonaws.licensemanager#String"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "FailedDependency",
        "httpResponseCode": 424
      },
      "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 424
    }
  },
  "com.amazonaws.licensemanager#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#FilterName",
        "traits": {
          "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
        }
      },
```

```
            "Values": {
              "target": "com.amazonaws.licensemanager#FilterValues",
              "traits": {
                "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n          describe operation. Filters can be used
to match a set of resources by specific criteria,\n          such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanager#FilterLimitExceededException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "FilterLimitExceeded",
              "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Filter"
          }
        },
        "com.amazonaws.licensemanager#FilterName": {
          "type": "string"
        },
        "com.amazonaws.licensemanager#FilterValue": {
          "type": "string"
        },
        "com.amazonaws.licensemanager#FilterValues": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#FilterValue",
            "traits": {
```

```
          "smithy.api#xmlName": "item"
        }
      }
    },
    "com.amazonaws.licensemanager#Filters": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Filter",
        "traits": {
          "smithy.api#xmlName": "item"
        }
      }
    },
    "com.amazonaws.licensemanager#GetAccessToken": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
 tokens\n      are valid for one hour.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetAccessTokenRequest": {
      "type": "structure",
      "members": {
        "Token": {
```

```
                "target": "com.amazonaws.licensemanager#TokenString",
                "traits": {
                  "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
                  "smithy.api#required": {}
                }
              }
            },
            "TokenProperties": {
              "target": "com.amazonaws.licensemanager#MaxSize3StringList",
              "traits": {
                "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
        "type": "structure",
        "members": {
          "AccessToken": {
            "target": "com.amazonaws.licensemanager#TokenString",
            "traits": {
              "smithy.api#documentation": "<p>Temporary access token.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      }
    },
    "com.amazonaws.licensemanager#GetGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
    }
  },
  "com.amazonaws.licensemanager#GetGrantRequest": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant version.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#GetGrantResponse": {
    "type": "structure",
    "members": {
      "Grant": {
        "target": "com.amazonaws.licensemanager#Grant",
        "traits": {
```

```
                    "smithy.api#documentation": "<p>Grant details.</p>"
                }
            }
        },
        "traits": {

"smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicense": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"

        },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#GetLicenseConfiguration": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
        },
        "output": {
```

```
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits":
{
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurationId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
```

```
          }
        },
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules.</p>"
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",

          "traits": {
            "smithy.api#documentation": "<p>Number of available licenses.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
```

```
          "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License configuration status.</p>"
        }
      }
    },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
        }
      },
      "ConsumedLicenseSummaryList": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
        }
      },
      "ManagedResourceSummaryList": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanager#TagList",
         "traits": {
          "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
        }
      },
      "ProductInformationList": {
        "target": "com.amazonaws.licensemanager#ProductInformationList",
        "traits": {
          "smithy.api#documentation": "<p>Product information.</p>"
        }
      },
      "AutomatedDiscoveryInformation": {
        "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
        "traits": {
          "smithy.api#documentation": "<p>Automated discovery information.</p>"
        }
      },
      "DisassociateWhenNotFound": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
```

```
            "traits": {
                "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#GetLicenseConversionTask": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {

"target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
    }
},
"com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
    "type": "structure",
    "members": {
        "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
                "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
```

```
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
      "type": "structure",
      "members": {
        "LicenseConversionTaskId": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
          "traits": {
            "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
          }
        },
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
          }
        },
        "SourceLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
          }
        },
        "DestinationLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
          "traits": {
            "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
          }
```

```
            },

    "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                }
            },
            "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

            },
            {
              "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
          "type": "structure",
          "members": {
            "LicenseManagerReportGeneratorArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
                "smithy.api#required":
{}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {
          "ReportGenerator": {
            "target": "com.amazonaws.licensemanager#ReportGenerator",
            "traits": {
              "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
```

```
    },
    "com.amazonaws.licensemanager#GetLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseResponse": {
      "type": "structure",
      "members": {
        "License": {
          "target": "com.amazonaws.licensemanager#License",
          "traits": {
            "smithy.api#documentation": "<p>License details.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },

    "com.amazonaws.licensemanager#GetLicenseUsage": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
      },
      "errors": [
        {
```

```
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
        "type": "structure",
        "members": {
          "LicenseUsage": {
            "target": "com.amazonaws.licensemanager#LicenseUsage",
            "traits": {
              "smithy.api#documentation": "<p>License usage details.</p>"
            }
          }
```

```
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#GetServiceSettings": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
            "type": "structure",
            "members": {
                "S3BucketArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"
```

```
                }
            },
            "SnsTopicArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
                }
            },
            "OrganizationConfiguration": {
                "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n        cross-account discovery.</p>"
                }
            },
            "EnableCrossAccountsDiscovery": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
                }

            },
            "LicenseManagerResourceShareArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n        provides member accounts with access to this share.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#Grant": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantName": {
```

```
            "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant name.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ParentArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Parent ARN.</p>",
                        "smithy.api#required": {}
                    }
                },
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>License ARN.</p>",
                        "smithy.api#required": {}
                    }
                },
                "GranteePrincipalArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {

"smithy.api#documentation": "<p>The grantee principal ARN.</p>",
                        "smithy.api#required": {}
                    }
                },
                "HomeRegion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                        "smithy.api#required": {}
                    }
                },
                "GrantStatus": {
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>",
                        "smithy.api#required": {}
                    }
                },
                "StatusReason": {
                    "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status reason.</p>"
                    }
                },
```

```json
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Granted operations.</p>",
            "smithy.api#required": {}
          }
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
            "smithy.api#documentation": "<p>The options specified for the grant.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Describes a grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#GrantList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Grant"
      }
    },
    "com.amazonaws.licensemanager#GrantStatus": {
      "type": "enum",
      "members": {
        "PENDING_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_WORKFLOW"
          }
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_ACCEPT"
          }
        },
        "REJECTED": {
```

```
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "REJECTED"

          }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits":
{
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
      "type": "structure",
      "members": {
```

```
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
      }
    },
    "LicenseName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>License name.</p>"
      }

},
    "ProductName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product name.</p>"
      }
    },
    "ProductSKU": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product SKU.</p>"
      }
    },
    "Issuer": {
      "target": "com.amazonaws.licensemanager#IssuerDetails",
      "traits": {
        "smithy.api#documentation": "<p>Granted license issuer.</p>"
      }
    },
    "HomeRegion": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanager#LicenseStatus",
      "traits": {
        "smithy.api#documentation": "<p>Granted license status.</p>"
      }
    },
    "Validity": {
      "target": "com.amazonaws.licensemanager#DatetimeRange",
      "traits": {
        "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
```

ISO8601-UTC format.</p>"
                }
            },
            "Beneficiary": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
                }
            },
            "Entitlements": {
                "target": "com.amazonaws.licensemanager#EntitlementList",
                "traits": {
                    "smithy.api#documentation": "<p>License entitlements.</p>"
                }
            },
            "ConsumptionConfiguration": {
                "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
                }
            },
            "LicenseMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Granted license metadata.</p>"
                }
            },
            "CreateTime": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

}
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Version of the granted license.</p>"
                }
            },
            "ReceivedMetadata": {
                "target": "com.amazonaws.licensemanager#ReceivedMetadata",
                "traits": {
                    "smithy.api#documentation": "<p>Granted license received metadata.</p>"
                }
            }
        },
        "traits": {

```
          "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
        }
      },
      "com.amazonaws.licensemanager#GrantedLicenseList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#GrantedLicense"
        }
      },
      "com.amazonaws.licensemanager#ISO8601DateTime":
 {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
        }
      },
      "com.amazonaws.licensemanager#Integer": {
        "type": "integer"
      },
      "com.amazonaws.licensemanager#InvalidParameterValueException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidParameterValueProvided",
            "httpResponseCode": 400
          },
          "smithy.api#documentation":
 "<p>One or more parameter values are not valid.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#InvalidResourceStateException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
```

```
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidResourceState",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
    },

    "com.amazonaws.licensemanager#InventoryFilter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the filter.</p>",
              "smithy.api#required": {}
            }
          },
          "Condition": {
            "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
            "traits": {
              "smithy.api#documentation": "<p>Condition of the filter.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }

    },
        "traits": {
          "smithy.api#documentation": "<p>An inventory filter.</p>"
        }
    },
    "com.amazonaws.licensemanager#InventoryFilterCondition": {
        "type": "enum",
        "members": {
          "EQUALS": {
            "target": "smithy.api#Unit",
```

```
          "traits": {
            "smithy.api#enumValue": "EQUALS"
          }
        },
        "NOT_EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "NOT_EQUALS"
          }
        },
        "BEGINS_WITH": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "BEGINS_WITH"
          }
        },
        "CONTAINS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CONTAINS"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#InventoryFilter"
      }
    },
    "com.amazonaws.licensemanager#Issuer": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Issuer name.</p>",
            "smithy.api#required": {}
          }
        },
        "SignKey": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
of sign and verify, \n      and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
          }
        }
```

```
          },
        "traits": {
          "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#IssuerDetails": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>"
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#License": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

      "smithy.api#documentation": "<p>License name.</p>"
            }
```

```
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#IssuerDetails",
          "traits": {
            "smithy.api#documentation": "<p>License issuer.</p>"
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {

"smithy.api#documentation": "<p>Home Region of the license.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        },
        "Entitlements": {
```

```
        "target": "com.amazonaws.licensemanager#EntitlementList",
              "traits": {
                 "smithy.api#documentation": "<p>License entitlements.</p>"
              }
           },
           "ConsumptionConfiguration": {
              "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
              "traits": {
                 "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
              }
           },
           "LicenseMetadata": {
              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                 "smithy.api#documentation": "<p>License metadata.</p>"
              }
           },
           "CreateTime": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                 "smithy.api#documentation": "<p>License
creation time.</p>"
              }
           },
           "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                 "smithy.api#documentation": "<p>License version.</p>"
              }
           }
        },
        "traits": {
           "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
        }
     },
     "com.amazonaws.licensemanager#LicenseConfiguration": {
        "type": "structure",
        "members": {
           "LicenseConfigurationId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                 "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
              }
           },
           "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
```

```
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules.</p>"
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
```

```
          "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Status of the license configuration.</p>"
        }
      },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

        }
      },
      "ConsumedLicenseSummaryList": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
        }
      },
      "ManagedResourceSummaryList": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
        }
      },
      "ProductInformationList": {
        "target": "com.amazonaws.licensemanager#ProductInformationList",
        "traits": {
          "smithy.api#documentation": "<p>Product information.</p>"
        }
      },
      "AutomatedDiscoveryInformation": {

        "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
        "traits": {
          "smithy.api#documentation": "<p>Automated discovery information.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n        consumed and enforced by License Manager. Components include specifications for
the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n        must be associated with a
host), and the number of licenses purchased and used.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
        "type": "structure",
        "members": {
          "ResourceArn":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of server resource.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
            }
          },
          "AssociationTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
            }
          },
          "AmiAssociationScope": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
```

```
          }
        },
        "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
          "type": "enum",
          "members":
{
            "AVAILABLE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "AVAILABLE"
              }
            },
            "DISABLED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "DISABLED"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
              }
            },
            "ResourceType": {
              "target": "com.amazonaws.licensemanager#ResourceType",
              "traits":
{
                "smithy.api#documentation": "<p>Type of resource.</p>"
              }
            },
            "ResourceStatus": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Status of the resource.</p>"
              }
            },
            "ResourceOwnerId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
              }
            },
```

```
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
          }
        },

    "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      }
    },
      "traits": {
        "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurations": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfiguration"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
      "members": {
        "UsageOperation": {
          "target": "com.amazonaws.licensemanager#UsageOperation",
          "traits": {
            "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n        </p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {

          "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
              }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the conversion task.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "StartTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
```

```
              "traits": {
                 "smithy.api#documentation":
"<p>The time the conversion task was started at.</p>"
              }
          },
          "LicenseConversionTime": {
             "target": "com.amazonaws.licensemanager#DateTime",
             "traits": {
                "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
             }
          },
          "EndTime": {
             "target": "com.amazonaws.licensemanager#DateTime",
             "traits": {
                "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
             }
          }
       },
       "traits": {
          "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
       }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskId": {
       "type": "string",
       "traits": {


      "smithy.api#length": {
            "min": 0,
            "max": 50
         },
         "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
       }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
       "type": "enum",
       "members": {
          "IN_PROGRESS": {
             "target": "smithy.api#Unit",
             "traits": {
                "smithy.api#enumValue": "IN_PROGRESS"
             }
          },
          "SUCCEEDED": {
             "target": "smithy.api#Unit",
             "traits": {
                "smithy.api#enumValue": "SUCCEEDED"
             }
```

```
                },
                "FAILED": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "FAILED"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#LicenseConversionTasks":
        {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTask"
            }
        },
        "com.amazonaws.licensemanager#LicenseCountingType": {
            "type": "enum",
            "members": {
                "VCPU": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "vCPU"
                    }
                },
                "INSTANCE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Instance"
                    }
                },
                "CORE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Core"
                    }
                },
                "SOCKET": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Socket"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#LicenseDeletionStatus": {
            "type": "enum",
            "members": {
```

```
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#License"
      }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailure": {

      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Resource type.</p>"
          }
        },
        "ErrorMessage": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Error message.</p>"
          }
        },
        "FailureTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {

"smithy.api#documentation": "<p>Failure time.</p>"
        }
```

```
            },
            "OperationName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Name of the operation.</p>"
              }
            },
            "ResourceOwnerId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
              }
            },
            "OperationRequestedBy": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
              }
            },
            "MetadataList": {

              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                "smithy.api#documentation": "<p>Reserved.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseOperationFailureList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecification": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
```

```
          "smithy.api#required": {}
          }
        },
        "AmiAssociationScope": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseSpecifications": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseSpecification"
      }
    },
    "com.amazonaws.licensemanager#LicenseStatus": {
      "type": "enum",
      "members": {
        "AVAILABLE": {
          "target": "smithy.api#Unit",

          "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "PENDING_AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_AVAILABLE"
          }
        },
        "DEACTIVATED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DEACTIVATED"
          }
        },
        "SUSPENDED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUSPENDED"
          }
```

```
      },
      "EXPIRED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "EXPIRED"
        }
      },

   "PENDING_DELETE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_DELETE"
        }
      },
      "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DELETED"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#LicenseUsage": {
    "type": "structure",
    "members": {
      "EntitlementUsages": {
        "target": "com.amazonaws.licensemanager#EntitlementUsageList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlement usages.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

    }
  },
  "com.amazonaws.licensemanager#LicenseUsageException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "LicenseUsageFailure",
        "httpResponseCode": 412
```

```
          },
          "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 412
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
        },
        "output": {

    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
```

```
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
                        "smithy.api#required": {}
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
            "type": "structure",
            "members": {
                "LicenseConfigurationAssociations": {
                    "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
                    "traits": {
                        "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
                    }
                },
                "NextToken": {

"target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListDistributedGrants": {
            "type": "operation",
            "input": {
```

          "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
        },
      "output": {
          "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
        },
      "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
           },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
      "traits": {
          "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
        "type": "structure",
        "members": {
          "GrantArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {

          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>LicenseArn</code>\n   </p>\n
        \n   \n   <p>\n   <code>GrantStatus</code>\n   </p>\n   \n

```
          <li>\n            <p>\n                <code>GranteePrincipalARN</code>\n            </p>\n        </li>\n
<li>\n            <p>\n                <code>ProductSKU</code>\n            </p>\n        </li>\n        <li>\n
<p>\n                <code>LicenseIssuerName</code>\n
        </p>\n        </li>\n    </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
            "type": "structure",
            "members": {
                "Grants": {
                    "target": "com.amazonaws.licensemanager#GrantList",

            "traits": {
                        "smithy.api#documentation": "<p>Distributed grant details.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
```

```
            },
        "output": {
            "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
        },
        "errors": [
            {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
        "type": "structure",
        "members": {
          "LicenseOperationFailureList": {
            "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
            "traits": {
              "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
        },

    "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
```

```
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configurations
 for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n        Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n        Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n
<p>\n        <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n        Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
   </li>\n        <li>\n        <p>\n        <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n        Logical operators are
```

```
          <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n      </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurations": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurations",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",

        "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
            }
          },
          "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
            "type": "structure",
            "members": {
              "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
              },
              "Filters": {
                "target": "com.amazonaws.licensemanager#Filters",
                "traits": {
                  "smithy.api#documentation": "<p>\n        Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n      </p>"
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
            "type": "structure",
            "members": {
              "LicenseConversionTasks": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
                "traits": {
                  "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
                }
              },
```

```
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target":
"com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
```

```
                }
            },
        "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
            "type": "structure",
            "members": {
                "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n            <ul>\n               <li>\n                  <p>\n                     <code>LicenseConfigurationArn</code>\n
</p>\n               </li>\n            </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits":
{
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
            "type": "structure",
            "members": {
                "ReportGenerators": {
                    "target": "com.amazonaws.licensemanager#ReportGeneratorList",
                    "traits": {
                        "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
```

```
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
```

```
              "traits": {

        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
        "type": "structure",
        "members": {
          "LicenseSpecifications": {
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
              "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseVersions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
```

```
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
```

```
              "target": "com.amazonaws.licensemanager#LicenseList",
              "traits": {
                 "smithy.api#documentation": "<p>License details.</p>"
              }
           },

"NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
           }
        },
        "traits": {
           "smithy.api#output": {}
        }
     },
     "com.amazonaws.licensemanager#ListLicenses": {
        "type": "operation",
        "input": {
           "target": "com.amazonaws.licensemanager#ListLicensesRequest"
        },
        "output": {
           "target": "com.amazonaws.licensemanager#ListLicensesResponse"
        },
        "errors": [
           {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
           },
           {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
           },
           {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
           },
           {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
           },
           {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
           },
           {
              "target": "com.amazonaws.licensemanager#ValidationException"
           }
        ],
        "traits": {
           "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
        }
```

```
        },
        "com.amazonaws.licensemanager#ListLicensesRequest": {
            "type": "structure",
            "members": {
                "LicenseArns": {
                    "target": "com.amazonaws.licensemanager#ArnList",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
                    }
                },

            "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n        <p>\n        <code>Beneficiary</code>\n        </p>\n
        </li>\n        <li>\n        <p>\n        <code>ProductSKU</code>\n        </p>\n        </li>\n
    <li>\n        <p>\n        <code>Fingerprint</code>\n        </p>\n        </li>\n        <li>\n
<p>\n        <code>Status</code>\n        </p>\n        </li>\n        </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token
 for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListLicensesResponse": {
            "type": "structure",
            "members": {
                "Licenses": {
                    "target": "com.amazonaws.licensemanager#LicenseList",
                    "traits": {
                        "smithy.api#documentation": "<p>License details.</p>"
                    }
                },
```

"NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {

"smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrants": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
         },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n   recipient as this Amazon Web Services account, your organization, or an organizational unit\n
   (OU) to which this member account belongs.</p>"
      }

```
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
            "smithy.api#required": {}
          }
        },
        "Filters": {
```

                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
        \n   \n   <p>\n   <code>ParentArn</code>\n   </p>\n   \n
\n   <p>\n   <code>GranteePrincipalArn</code>\n   </p>\n   \n   "
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
        "type": "structure",
        "members": {
            "Grants": {
                "target": "com.amazonaws.licensemanager#GrantList",
                "traits": {
                    "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
        "type": "structure",

```
"members": {
  "GrantArns": {
    "target": "com.amazonaws.licensemanager#ArnList",
     "traits": {
       "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
     }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanager#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n      <ul>\n         <li>\n            <p>\n               <code>ProductSKU</code>\n            </p>\n
</li>\n         <li>\n            <p>\n               <code>LicenseIssuerName</code>\n            </p>\n
</li>\n         <li>\n            <p>\n               <code>LicenseArn</code>\n            </p>\n         </li>\n
<li>\n            <p>\n               <code>GrantStatus</code>\n            </p>\n         </li>\n         <li>\n
<p>\n
      <code>GranterAccountId</code>\n            </p>\n         </li>\n      </ul>"
    }
  },
  "NextToken": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
      "smithy.api#documentation": "<p>Token for the next set of results.</p>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanager#MaxSize100",
    "traits": {
      "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
  "type": "structure",
  "members": {
    "Grants": {
      "target": "com.amazonaws.licensemanager#GrantList",
       "traits": {
         "smithy.api#documentation": "<p>Received grant details.</p>"
       }
    },
    "NextToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
```

```
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#ListReceivedLicenses": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists received licenses.</p>"
  }
},
"com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
{
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
```

```
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {

    "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>Beneficiary</code>\n            </p>\n
          </li>\n          <li>\n            <p>\n              <code>ProductSKU</code>\n
        </p>\n          </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
```

```
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#GrantedLicenseList",
            "traits": {
              "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
        "type": "structure",
        "members": {
          "LicenseArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
            }
          },

          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>ProductSKU</code>\n            </p>\n
```

```
           </li>\n        <li>\n        <p>\n          <code>Status</code>\n        </p>\n      </li>\n
<li>\n       <p>\n          <code>Fingerprint</code>\n          </p>\n      </li>\n      <li>\n
<p>\n          <code>IssuerName</code>\n          </p>\n      </li>\n      <li>\n        <p>\n
<code>Beneficiary</code>\n          </p>\n      </li>\n      </ul>"
                }
              },
              "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                  "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
            "type": "structure",
            "members": {
              "Licenses": {
                "target": "com.amazonaws.licensemanager#GrantedLicenseList",
                "traits": {
                  "smithy.api#documentation": "<p>Received license details.</p>"
                }
              },

              "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#ListResourceInventory": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
```

          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#FailedDependencyException"
            },
            {
              "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#InventoryFilterList",

```
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n      are supported:</p>\n    <ul>\n       <li>\n          <p>\n             <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
    </li>\n       <li>\n          <p>\n             <code>application_name</code> - The name of the
application. \n          Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
   </li>\n       <li>\n          <p>\n             <code>license_included</code> - The type of license included. \n
     Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n          Possible values
are <code>sql-server-enterprise</code> | \n          <code>sql-server-standard</code> | \n          <code>sql-
server-web</code> | \n          <code>windows-server-datacenter</code>.</p>\n       </li>\n       <li>\n
   <p>\n             <code>platform</code> - The platform of the resource.\n          Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n       </li>\n       <li>\n          <p>\n
     <code>resource_id</code> - The ID of the resource. \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n       </li>\n       <li>\n          <p>\n
   <code>tag:<key></code> - The key/value combination of a tag assigned \n          to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n          <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n       </li>\n    </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
        "type": "structure",
        "members": {
            "ResourceInventoryList": {
                "target": "com.amazonaws.licensemanager#ResourceInventoryList",
                "traits": {
                    "smithy.api#documentation": "<p>Information
about the resources.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListTagsForResource": {
        "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

      "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
```

```
                    "target": "com.amazonaws.licensemanager#TagList",
                    "traits": {
                        "smithy.api#documentation": "<p>Information about the tags.</p>"
                    }
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListTokens": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListTokensRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListTokensResponse"
        },
        "errors":
[
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists your tokens.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListTokensRequest": {
        "type": "structure",
        "members": {
            "TokenIds": {
                "target": "com.amazonaws.licensemanager#StringList",
                "traits": {
```

                    "smithy.api#documentation": "<p>Token IDs.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    \n   \n   <p>\n   <code>LicenseArns</code>\n   </p>\n   \n
"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
            "Tokens": {
                "target": "com.amazonaws.licensemanager#TokenList",
                "traits": {
                    "smithy.api#documentation": "<p>Received token details.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }

                    "smithy.api#documentation": "<p>Token IDs.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    \n   \n   <p>\n   <code>LicenseArns</code>\n   </p>\n   \n
"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
            "Tokens": {
                "target": "com.amazonaws.licensemanager#TokenList",
                "traits": {
                    "smithy.api#documentation": "<p>Received token details.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }

```
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying license\n        consumption details by resource at a selected point in time. Use this action to audit the\n        current license consumption for any license inventory and configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n      are supported:</p>\n      <ul>\n         <li>\n            <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
    Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n         </li>\n
 <li>\n            <p>\n               <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n         Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n         </li>\n         <li>\n            <p>\n
  <code>resourceAccount</code> - The ID of the account that owns the resource. \n         Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n         </li>\n      </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {

  "LicenseConfigurationUsageList": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
```

```
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Location": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Long": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#ManagedResourceSummary": {
      "type": "structure",
      "members": {
        "ResourceType":
{
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
          }
        },
        "AssociationCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
      }
    },
    "com.amazonaws.licensemanager#MaxSize100": {

      "type": "integer",
      "traits": {
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
```

```
"com.amazonaws.licensemanager#MaxSize3StringList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 3
        }
    }
},
"com.amazonaws.licensemanager#Message": {
    "type": "string"
},
"com.amazonaws.licensemanager#Metadata": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The key name.</p>"
            }
        },

        "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
    }
},
"com.amazonaws.licensemanager#MetadataList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#Metadata"
    }
},
"com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#Options": {
        "type": "structure",
        "members": {
          "ActivationOverrideBehavior": {
            "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
            "traits": {
              "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n        Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n        <ul>\n        <li>\n          <p>As a license administrator, you can
optionally specify an\n          <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
</li>\n        <li>\n          <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n          account in your
organization.</p>\n        </li>\n        <li>\n          <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n          <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n          are activating the grant.</p>\n        </li>\n        </ul>\n        <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n        <dd>\n          <p>Use this value to activate a grant
without replacing any member accounts\n          active grants for the same product.</p>\n        </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n        <dd>\n          <p>Use this value to activate a
grant and disable other active grants in any\n          member accounts
 for the same product. This action will also replace their\n          previously activated grants with this activated
grant.</p>\n        </dd>\n        </dl>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n        override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
      },
      "com.amazonaws.licensemanager#OrganizationConfiguration": {
        "type": "structure",
        "members": {
          "EnableIntegration": {
            "target": "com.amazonaws.licensemanager#Boolean",
            "traits": {
              "smithy.api#default": false,
              "smithy.api#documentation":
```

```
          "<p>Enables Organizations integration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
        }
    },
    "com.amazonaws.licensemanager#PrincipalArnList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#Arn"
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 1
            }
        }
    },
    "com.amazonaws.licensemanager#ProductInformation": {
        "type": "structure",
        "members": {
            "ResourceType": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Resource type. The
possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductInformationFilterList": {
                "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
                "traits": {
                    "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n        <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n        <p>The following filters and are supported when
the resource type \n        is <code>SSM_MANAGED</code>:</p>\n
        <ul>\n        <li>\n            <p>\n                <code>Application Name</code> - The name of the
application.\n            Logical operator is <code>EQUALS</code>.</p>\n        </li>\n        <li>\n
<p>\n                <code>Application Publisher</code> - The publisher of the application.\n            Logical
operator is <code>EQUALS</code>.</p>\n        </li>\n        <li>\n            <p>\n
<code>Application Version</code> - The version of the application.\n            Logical operator is
<code>EQUALS</code>.</p>\n        </li>\n        <li>\n            <p>\n                <code>Platform
```

Name</code> - The name of the platform.\n   Logical operator is <code>EQUALS</code>.</p>\n
\n   \n   <p>\n   <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
        \n   \n   <p>\n   <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>. The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n   </p>\n   \n   \n   <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n   Logical operator is <code>NOT_EQUALS</code>.</p>\n   \n
\n   <p>\n   <code>License Included</code> - The type of license included.\n   Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n   Possible values are:
 <code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-server-
web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n
<p>The following filters and logical operators are supported when the resource type\n   is
<code>RDS</code>:</p>\n   \n   \n   <p>\n   <code>Engine Edition</code> -
The edition of the database engine.\n   Logical operator is <code>EQUALS</code>.\n   Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n   \n   \n   <p>\n   <code>License Pack</code> - The license
pack.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>data
guard</code> | \n   <code>diagnostic pack sqlt</code> | \n
        <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n   <code>olap</code>.</p>\n
  \n   ",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#ProductInformationFilter": {
          "type": "structure",
          "members": {
            "ProductInformationFilterName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Filter name.</p>",
                "smithy.api#required": {}
              }
            },
            "ProductInformationFilterValue": {
              "target": "com.amazonaws.licensemanager#StringList",

              "traits": {
                  "smithy.api#documentation": "<p>Filter value.</p>"
              }
            },

```
      "ProductInformationFilterComparator": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Logical operator.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes product information filters.</p>"
    }
  },
  "com.amazonaws.licensemanager#ProductInformationFilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ProductInformationFilter"
    }
  },
  "com.amazonaws.licensemanager#ProductInformationList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ProductInformation"
    }
  },
  "com.amazonaws.licensemanager#ProvisionalConfiguration": {
    "type": "structure",
    "members": {
      "MaxTimeToLiveInMinutes": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#RateLimitExceededException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
```

```
          "aws.protocols#awsQueryError": {
            "code": "RateLimitExceeded",
            "httpResponseCode": 429
          },
          "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 429
        }
      },
      "com.amazonaws.licensemanager#ReceivedMetadata": {
        "type": "structure",
        "members": {
          "ReceivedStatus": {
            "target": "com.amazonaws.licensemanager#ReceivedStatus",
            "traits": {
              "smithy.api#documentation": "<p>Received status.</p>"
            }
          },
          "ReceivedStatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>Received
status reason.</p>"
            }
          },
          "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed operations.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
        }
      },
      "com.amazonaws.licensemanager#ReceivedStatus": {
        "type": "enum",
        "members": {
          "PENDING_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_WORKFLOW"
            }
          },
          "PENDING_ACCEPT": {
            "target": "smithy.api#Unit",
```

```
                "traits": {
                    "smithy.api#enumValue":
"PENDING_ACCEPT"
                }
            },
            "REJECTED": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "REJECTED"
                }
            },
            "ACTIVE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "ACTIVE"
                }
            },
            "FAILED_WORKFLOW": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "FAILED_WORKFLOW"
                }
            },
            "DELETED": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DELETED"
                }
            },
            "DISABLED": {
                "target": "smithy.api#Unit",
                 "traits": {
                    "smithy.api#enumValue": "DISABLED"
                }
            },
            "WORKFLOW_COMPLETED": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "WORKFLOW_COMPLETED"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#RedirectException": {
        "type": "structure",
        "members": {
            "Location": {
                "target": "com.amazonaws.licensemanager#Location",
```

```
          "traits": {
              "smithy.api#httpHeader": "Location"
          }
        },
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
         "smithy.api#error": "client",
        "smithy.api#httpError": 308
      }
    },
    "com.amazonaws.licensemanager#RejectGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#RejectGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#RejectGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
```

```
              }
          },
        "com.amazonaws.licensemanager#RejectGrantRequest": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}

    }
        },
        "com.amazonaws.licensemanager#RejectGrantResponse": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant ARN.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant version.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#RenewType":
{
            "type": "enum",
            "members": {
```

```
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "WEEKLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Weekly"
          }
        },
        "MONTHLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Monthly"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportContext": {
      "type": "structure",
      "members": {
        "licenseConfigurationArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {

      "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReportFrequency": {
      "type": "structure",
      "members": {
        "value": {
          "target": "com.amazonaws.licensemanager#Integer",
          "traits": {
            "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
          }
        },
        "period": {
```

```
              "target": "com.amazonaws.licensemanager#ReportFrequencyType",

          "traits": {
              "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReportFrequencyType": {
      "type": "enum",
      "members": {
        "DAY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DAY"
          }
        },
        "WEEK": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WEEK"
          }
        },
        "MONTH": {
          "target": "smithy.api#Unit",
          "traits":
{

            "smithy.api#enumValue": "MONTH"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportGenerator": {
      "type": "structure",
      "members": {
        "ReportGeneratorName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>"
          }
        },
        "ReportType": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
```

```
            "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
          }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
          }
        },
        "ReportFrequency": {
          "target": "com.amazonaws.licensemanager#ReportFrequency",
          "traits": {
            "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
          }
        },
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
          }
        },
        "LastRunStatus": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
          }
        },
        "LastRunFailureReason":
{
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
          }
        },
        "LastReportGenerationTime": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
          }
        },
        "ReportCreatorAccount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
          }
        },
```

```
        "Description": {
            "target": "com.amazonaws.licensemanager#String",

    "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
        },
        "S3Location": {
          "target": "com.amazonaws.licensemanager#S3Location",
          "traits": {
            "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
          }
        },
        "CreateTime": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the report was created.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
          }

    }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportGenerator"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorName": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "com.amazonaws.licensemanager#ReportType": {
```

```
      "type": "enum",
      "members": {
        "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
          }

      },
        "LICENSE_CONFIGURATION_USAGE_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationUsageReport"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportTypeList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportType"
      }
    },
    "com.amazonaws.licensemanager#ResourceInventory": {
      "type": "structure",
      "members": {
        "ResourceId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the resource.</p>"
          }
        },
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits":

{

            "smithy.api#documentation": "<p>Type of resource.</p>"
          }
        },
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },
        "Platform": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>Platform of the resource.</p>"
                }
            },
            "PlatformVersion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
                }
            },
            "ResourceOwningAccountId":
{
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about a resource.</p>"
        }
    },
    "com.amazonaws.licensemanager#ResourceInventoryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#ResourceInventory"
        }
    },
    "com.amazonaws.licensemanager#ResourceLimitExceededException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "ResourceLimitExceeded",

                "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#ResourceNotFoundException": {
        "type": "structure",
        "members": {
```

```
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InvalidResource.NotFound",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>The resource cannot be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#ResourceType": {
      "type": "enum",
      "members": {
        "EC2_INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_INSTANCE"
          }
        },
        "EC2_HOST": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_HOST"
          }
        },
        "EC2_AMI": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_AMI"
          }
        },
        "RDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "RDS"
          }
        },
        "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
          }
        }
      }
```

```
        },
        "com.amazonaws.licensemanager#S3Location": {
            "type": "structure",
            "members": {
                "bucket": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
                    }
                },
                "keyPrefix": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
            }
        },
        "com.amazonaws.licensemanager#ServerInternalException": {
            "type": "structure",
            "members": {
                "Message": {
                    "target": "com.amazonaws.licensemanager#Message"
                }
            },
            "traits": {
                "aws.protocols#awsQueryError": {
                    "code": "InternalError",
                    "httpResponseCode": 500
                },
                "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
                "smithy.api#error": "server",
                "smithy.api#httpError": 500
            }
        },
        "com.amazonaws.licensemanager#SignedToken": {
            "type": "string",
            "traits": {
                "smithy.api#length": {
                    "min": 4096
                }
            }
        },
        "com.amazonaws.licensemanager#StatusReasonMessage": {
```

```
        "type": "string",
        "traits":
{
        "smithy.api#length": {
           "min": 0,
           "max": 400
        },
        "smithy.api#pattern": "^[\\s\\S]+$"
     }
   },
   "com.amazonaws.licensemanager#String": {
      "type": "string"
   },
   "com.amazonaws.licensemanager#StringList": {
      "type": "list",
      "member": {
         "target": "com.amazonaws.licensemanager#String"
      }
   },
   "com.amazonaws.licensemanager#Tag": {
      "type": "structure",
      "members": {
         "Key": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
               "smithy.api#documentation": "<p>Tag key.</p>"
            }
         },
         "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
               "smithy.api#documentation":
"<p>Tag value.</p>"
            }
         }
      },
      "traits": {
         "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
      }
   },
   "com.amazonaws.licensemanager#TagKeyList": {
      "type": "list",
      "member": {
         "target": "com.amazonaws.licensemanager#String"
      }
   },
   "com.amazonaws.licensemanager#TagList": {
      "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanager#Tag"
        }
    },
    "com.amazonaws.licensemanager#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#TagResourceResponse"
      },
      "errors": [
        {
          "target":
"com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#TagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits":
{
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
```

```
      "traits": {
        "smithy.api#documentation": "<p>One or more tags.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanager#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#TokenData": {
  "type": "structure",
  "members": {
    "TokenId": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token ID.</p>"
      }
    },
    "TokenType": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
      }
    },
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
      }
    },
    "ExpirationTime": {
      "target": "com.amazonaws.licensemanager#ISO8601DateTime",
      "traits": {
        "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
format.</p>"
      }
    },
    "TokenProperties": {
      "target": "com.amazonaws.licensemanager#MaxSize3StringList",
```

```
        "traits": {
          "smithy.api#documentation": "<p>Data specified by the caller.</p>"
        }
      },
      "RoleArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
        }
      }
    },
    "traits": {

      "smithy.api#documentation": "<p>Describes a token.</p>"
    }
  },
  "com.amazonaws.licensemanager#TokenList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#TokenData"
    }
  },
  "com.amazonaws.licensemanager#TokenString": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 4096
      },
      "smithy.api#pattern": "^\\S+$"
    }
  },
  "com.amazonaws.licensemanager#TokenType": {
    "type": "enum",
    "members": {
      "REFRESH_TOKEN": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "REFRESH_TOKEN"
        }
```

```
        }
      }
    },
    "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
{
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
         },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
      }
    },
```

```
"com.amazonaws.licensemanager#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licensemanager#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
    },
    "output": {

"target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "LicenseConfigurationStatus": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
              "traits": {
                "smithy.api#documentation": "<p>New status of the license configuration.</p>"
              }
            },
            "LicenseRules": {
              "target": "com.amazonaws.licensemanager#StringList",
              "traits": {

    "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
    configuration is licenseAffinityToHost.</p>"
              }
            },
            "LicenseCount": {
              "target": "com.amazonaws.licensemanager#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
              }
            },
            "LicenseCountHardLimit": {
```

                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
                }
            },
            "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
                }
            },
            "Description": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>New description of the license configuration.</p>"
                }
            },
            "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                    "smithy.api#documentation": "<p>New product information.</p>"
                }
            },
            "DisassociateWhenNotFound": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            }
        },
        "traits":
{
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportGeneratorName": {
```

```
            "target": "com.amazonaws.licensemanager#ReportGeneratorName",
            "traits": {
              "smithy.api#documentation": "<p>Name of the report generator.</p>",
              "smithy.api#required": {}
            }
          },
          "Type": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
              "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n        <ul>\n          <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n          </li>\n          <li>\n            <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n          </li>\n          </ul>",
              "smithy.api#required": {}
            }
          },
          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>The report context.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits":
{
              "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientRequestToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
          }
        },
        "traits": {
```

```
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
 {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },

          {
            "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
          },
          {
            "target": "com.amazonaws.licensemanager#LicenseUsageException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
```

```
        }
      },

    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
            "smithy.api#required": {}
          }
        },
        "AddLicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
          }
        },
        "RemoveLicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation":
  "<p>ARNs of the license configurations to remove.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
      },
      "errors": [
```

```
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
       },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
    }
  },
  "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "S3BucketArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
        }
      },
      "SnsTopicArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
        }
      },
      "OrganizationConfiguration": {
        "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
        }
      },
      "EnableCrossAccountsDiscovery": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
```

```
                    "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
      "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#UsageOperation": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          }
        }
      },
      "com.amazonaws.licensemanager#ValidationException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

      "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      }
    }
  }
}
```

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```go
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version
 of the license.
 //
 // This member is required.
 SourceVersion *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus
```

```go
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack,
options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
```

```go
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "DeleteLicense",
	}
}
```

1.29 aws-aws-sdk-go-v2 1.10.7

1.29.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil,
 err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType
```

```go
    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err !=
    nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
```

```
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack);
 err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
```

```go
 return err
 }
 if err = addDisableHTTPSMiddleware(stack,
 options); err != nil {
 return err
 }
 return nil
}


func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
```
AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

name: License Scan

on: [pull_request]

jobs:
 licensescan:
  name: License Scan
  runs-on: ubuntu-latest
  strategy:
   matrix:
    python-version: [3.9]

  steps:
   - name: Checkout target
    uses: actions/checkout@v2
    with:
     path: sdkbase
     ref: ${{ github.base_ref }}
   - name: Checkout this ref
    uses: actions/checkout@v2
    with:
     path: new-ref
     fetch-depth: 0
   - name: Get Diff
    run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }} ${{ github.sha }} > refDiffFiles.txt
   - name: Get Target Files
    run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

```yaml
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options)) (*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns, c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```go
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary
    *string

    noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
 err != nil {
 return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return
 err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
 middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
```

```
    return err
   }
   if err = addResponseErrorMiddleware(stack); err != nil {
    return err
   }
   if err = addRequestResponseLogging(stack, options); err != nil {
    return err
   }
   if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
   }
   return nil
  }

  func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
  {
   return &awsmiddleware.RegisterServiceMetadata{
    Region:        region,
    ServiceID:     ServiceID,
    OperationName: "CheckInLicense",
   }
  }
```

```
  {
    "smithy": "2.0",
    "shapes": {
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
```

```
              "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n     Filters can be used to match a set of resources by specific criteria.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {

            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/GetServiceSettings"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
```

```
{
    "type": "structure",
    "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n      Organizations integration has been enabled, and
which Regions data will be aggregated from.</p>"
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays
the aggregated data for Linux\n      subscriptions.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
        "AmiId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }

        },
          "AccountID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
            }
          },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
```

```
                      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n          operation values</a> in the <i>License Manager User Guide</i> .</p>"
                }
            },
            "LastUpdatedTime": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
                }
            },
            "SubscriptionName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
```

```
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
                }
            ],

            "traits": {
                "aws.api#service": {
                    "sdkId": "License Manager Linux Subscriptions",
                    "arnNamespace": "license-manager-linux-subscriptions",
                    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
                },
                "aws.auth#sigv4": {
                    "name": "license-manager-linux-subscriptions"
                },
                "aws.protocols#restJson1": {},
                "smithy.api#cors": {
                    "additionalAllowedHeaders": [
                        "*",
                        "content-type",
                        "x-amz-content-sha256",
                        "x-amz-user-agent",
                        "x-amzn-platform-id",
                        "x-amzn-trace-id",
                        "content-length",
                        "x-api-key",
                        "authorization",
                        "x-amz-date",

 "x-amz-security-token",
                        "Access-Control-Allow-Headers",
                        "Access-Control-Allow-Methods",
                        "Access-Control-Allow-Origin"
                    ],
                    "additionalExposedHeaders": [
                        "x-amzn-errortype",
                        "x-amzn-requestid",
                        "x-amzn-trace-id"
                    ]
                },
                "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
```

```
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The
AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
              "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "type": "tree",
```

```
"rules": [
    {

  "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
              {
                  "ref": "UseFIPS"
              },
              true
            ]
        }
    ],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "type": "tree",
    "rules": [
        {

  "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                  {
                      "ref": "UseDualStack"
                  },
                  true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],

"endpoint": {
                "url": {
                    "ref": "Endpoint"
                },
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
```

```
                    }
                ]
            }
        ]
    },
    {
        "conditions": [],
        "type": "tree",
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",

                        "argv": [
                            {
                                "ref": "Region"
                            }
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "aws.partition",
                                "argv": [
                                    {
                                        "ref": "Region"
                                    }
                                ],

        "assign": "PartitionResult"
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
                                            {
                                                "ref": "UseFIPS"
                                            },
                                            true
                                        ]
```

```json
                },
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {

                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]

                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },

                                        "supportsDualStack"
                                    ]
                                }
                            ]
```

```
                                    }
                                ],
                                "type": "tree",
                                "rules": [
                                    {
                                        "conditions": [],
                                        "type": "tree",
                                        "rules": [
                                            {

                    "conditions": [],
                                                "endpoint": {
                                                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                    "properties": {},
                                                    "headers": {}
                                                },
                                                "type": "endpoint"
                                            }
                                        ]
                                    }
                                ]

                            },
                            {
                                "conditions": [],
                                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                                "type": "error"
                            }
                        ]
                    },
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {

                "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {
```

```
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,


{
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      }
                    ],
                    "type": "tree",

                    "rules": [
                      {
                        "conditions": [],
                        "type": "tree",
                        "rules": [
                          {
                            "conditions": [],
                            "endpoint": {
                              "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                              "properties": {},
                              "headers": {}

                            },
                            "type": "endpoint"
                          }
                        ]
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"
                  }
                ]
```

                },
                                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {

                        "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                            ]
                                        }
                                    ]
                                }
                            ],
                            "type": "tree",
                            "rules": [
                                {
                                    "conditions": [],
                                    "type": "tree",
                                    "rules": [
                                        {
                                            "conditions": [],

                                    "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}

                        },
                          "type": "endpoint"
                      }
                    ]
                  }
                ]
              },

  {
                          "conditions": [],
                          "error": "DualStack is enabled but this partition does not support DualStack",
                          "type": "error"
                      }
                    ]
                  },
                  {
                      "conditions": [],
                      "type": "tree",
                      "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                              "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                              },
                              "type": "endpoint"
                          }
                        ]
                    }
                  ]
              }
            ]
          },
          {
              "conditions": [],
              "error": "Invalid Configuration: Missing Region",
              "type": "error"
          }
        ]
      }
    ]
  },
  "smithy.rules#endpointTests": {
    "testCases": [
      {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
         "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {

              "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
             "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint":
```

```
{
                                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                            }
                        },
                        "params": {
                            "Region": "cn-north-1",
                            "UseFIPS": false,
                            "UseDualStack": false
                        }
                    },
                    {
                        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                            }
                        },
                        "params": {
                            "Region": "us-gov-east-1",
                            "UseFIPS": true,
                            "UseDualStack": true
                        }
                    },
                    {
                        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
                            }
                        },
                        "params": {
                            "Region": "us-gov-east-1",
                            "UseFIPS": true,
                            "UseDualStack": false
                        }
                    },
                    {
                        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                        "expect": {
                            "endpoint":
{

                                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
                            }
                        },
                        "params": {
                            "Region": "us-gov-east-1",
                            "UseFIPS": false,
                            "UseDualStack": true
```

```
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
               "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": true
```

```
                    }
                },
                {


    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
                    }
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {

                 "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled but this partition does not support DualStack"
                },
                "params": {
```

```
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",

    "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",

      "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
```

```
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,

   "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "Missing region",
          "expect": {
            "error": "Invalid Configuration: Missing Region"
          }

   }
      ],
      "version": "1.0"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
```

```
              "value": "Disabled",
              "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
        "type": "structure",
        "members": {
          "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
              "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type":
"operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with
commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {

    "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Instances"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by
 the name of\n      <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n      <p>The valid names for this filter
are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>AmiID</code>\n          </p>\n
</li>\n        <li>\n          <p>\n            <code>InstanceID</code>\n          </p>\n        </li>\n
<li>\n          <p>\n            <code>AccountID</code>\n          </p>\n        </li>\n        <li>\n
<p>\n            <code>Status</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>Region</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>UsageOperation</code>\n          </p>\n        </li>\n        <li>\n
  <p>\n            <code>ProductCode</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
  <code>InstanceType</code>\n          </p>\n        </li>\n      </ul>\n      <p>The valid Operators for this
filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>contains</code>\n          </p>\n
</li>\n        <li>\n          <p>\n            <code>equals</code>\n          </p>\n        </li>\n        <li>\n
    <p>\n            <code>Notequal</code>\n          </p>\n        </li>\n      </ul>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
```

```
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
          "type": "structure",
          "members": {
            "Instances": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
              "traits": {
                "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
```

```
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n     organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Subscriptions"
      }

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n     partially match, or don't
match a certain subscription's name.</p>\n     <p>The valid names for this filter are:</p>\n     <ul>\n
<li>\n          <p>\n               <code>Subscription</code>\n          </p>\n     </li>\n     </ul>\n
<p>The valid Operators for this filter are:</p>\n
     <ul>\n       <li>\n          <p>\n               <code>contains</code>\n          </p>\n     </li>\n
<li>\n          <p>\n               <code>equals</code>\n          </p>\n     </li>\n     <li>\n          <p>\n
     <code>Notequal</code>\n          </p>\n     </li>\n     </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
```

```
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>",
            "smithy.api#length": {


              "min": 1,
                "max": 16384
              }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "EQUAL",
            "value": "Equal",
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
```

```
              "documentation": "Not equal operator"
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
          "smithy.api#length": {
            "min": 1,
            "max": 20
          }
        }
      },

      "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled OrganizationIntegration"
            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled OrganizationIntegration"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [

{
              "name": "InProgress",
              "value": "InProgress",
              "documentation": "InProgress status"
            },
```

```
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits":
{
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
```

```
            "type": "structure",
          "members": {
            "Name": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"

            }
          },
            "Type": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
              }
            },
            "InstanceCount": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },

    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required": {}
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>",
```

```
      "smithy.api#required": {}
            }
          },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n     allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
        }
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },

    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
      },
      "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
          "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
        }
      },
      "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
```

```
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
          }
        }
      }
    }
}
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
```

```go
	out.ResultMetadata = metadata
	return out, nil
}

type ListLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//   - Beneficiary
	//   - ProductSKU
	//   - Fingerprint
	//
	//   - Status
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListLicensesOutput struct {

	// License details.
	Licenses []types.License

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
```

```go
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if
err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
```

```go
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
     Region:       region,
     ServiceID:
       ServiceID,
     OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
```

```go
	if params == nil {
		params = &DisassociateLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*DisassociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type DisassociateLicenseInput struct {

	// The type of license to remove from the workspace.
	//
	//
	This member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to remove the Grafana Enterprise license from.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

	// A structure containing information about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
```

```go
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "DisassociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
```

```go
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
 {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "DisassociateLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```go
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
	if params == nil {
		params = &CheckoutLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*CheckoutLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type
	CheckoutLicenseInput struct {

	// Checkout type.
	//
	// This member is required.
	CheckoutType types.CheckoutType

	// Unique, case-sensitive identifier that you provide to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// License entitlements.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Key fingerprint identifying the license.
	//
	// This member is required.
```

```
	KeyFingerprint *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// License beneficiary.
	Beneficiary *string

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

	// Checkout type.
	CheckoutType types.CheckoutType

	// Allowed license entitlements.
	EntitlementsAllowed []types.EntitlementData

	// Date and time at which the license checkout expires.
	Expiration *string

	// Date and time at which the license checkout is issued.
	IssuedAt *string

	// Amazon Resource
	Name (ARN) of the checkout license.
	LicenseArn *string

	// License consumption token.
	LicenseConsumptionToken *string

	// Node ID.
	NodeId *string

	// Signed token.
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil
 {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
```

```go
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack);
	err != nil {
		return err
	}
	if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
	middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
```

```go
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier
    that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see RFC 7518 Digital Signature with
    // RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData
```

```go
// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}
```

```go
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
 return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
 return
err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return
```

```go
    err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```go
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide
 to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration
```

```go
	// License entitlements.
	//
	// This member is required.
	Entitlements []types.Entitlement

	// Home Region for the license.
	//
	// This member is required.
	HomeRegion *string

	// License issuer.
	//
	// This member is required.
	Issuer *types.Issuer

	// License name.
	//
	// This member is required.
	LicenseName *string

	// Product name.
	//
	// This member is required.
	ProductName *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// Date and time range during which the license is valid, in ISO8601-UTC format.
	//
	// This member is required.
	Validity
	*types.DatetimeRange

	// Information about the license.
	LicenseMetadata []types.Metadata

	noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string
```

```go
	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
		return fmt.Errorf("add
protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
```

```
      return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
      return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
      return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
      return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
      return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
    err != nil {
      return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
      return err
    }
    if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
      return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
      return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
      return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
      return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
      return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
      return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
      return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
```

```go
  Region:        region,
  ServiceID:      ServiceID,
  OperationName:
"CreateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version
```

```go
    *string

    noSmithyDocumentSerde
}

type GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return
    err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
```

```go
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
    return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack);
err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
return nil
```

```
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "GetLicense",
 }
}
```

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
```

```go
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
   - ProductSKU
 //   - Status
 //   - Fingerprint
 //   - IssuerName
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense
```

```go
	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
```

```go
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack);
    err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
    middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
    string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
    Region:       region,
    ServiceID:    ServiceID,
    OperationName: "ListReceivedLicenses",
```

```
        }
    }
```

                    Apache License
                Version 2.0, January 2004
                http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership
   of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or Object
      form, that is based on (or derived from) the Work and for which the
      editorial
   revisions, annotations, elaborations, or other modifications
      represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding
those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions

 for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

 5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":

"<p>Details about an Active Directory identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p>\n   <note>\n   <p>Your estimated bill for charges on the number of users and related costs will take 48\n   hours to appear for billing periods that haven't closed (marked as Pending billing status) in Amazon Web Services Billing. For more information, see Viewing your\n   monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p>\n   </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/AssociateUser"
        },
        "smithy.api#idempotent": {}
      }

```json
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n    resource.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
```

subscriptions.</p>",
              "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
              },
              "smithy.api#idempotent": {}
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }

      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },

    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
```

```
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used
 to return more specific results from a describe\n     operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n     attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
```

```
        "traits": {
            "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n    settings such as the subnets to provision VPC endpoints.</p>",
            "smithy.api#required":
{}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of an identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
```

```
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status
of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
                    "smithy.api#required": {}
                }
            },
            "LastStatusCheckDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                }
            }

        }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
```

```
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits":
{
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
          "smithy.api#required": {}
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the user.</p>"
        }
      },
      "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
      },
      "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
```

```
          }
        }
    },
      "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",

      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
            }

        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
            }
        ],
        "traits": {
            "aws.api#service": {
                "sdkId": "License Manager User Subscriptions",
                "arnNamespace": "license-manager-user-subscriptions",
                "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
            },
            "aws.auth#sigv4": {
                "name": "license-manager-user-subscriptions"
             },
            "aws.protocols#restJson1": { },
            "smithy.api#cors": {
                "additionalAllowedHeaders": [
                    "*",
                    "content-type",
                    "x-amz-content-sha256",
                    "x-amz-user-agent",
                    "x-amzn-platform-id",
                    "x-amzn-trace-id",
                    "content-length",
                    "x-api-key",
                    "authorization",
                    "x-amz-date",
                    "x-amz-security-token",
                    "Access-Control-Allow-Headers",
                    "Access-Control-Allow-Methods",
                    "Access-Control-Allow-Origin"
                ],
                "additionalExposedHeaders": [
                    "x-amzn-errortype",
```

```
              "x-amzn-requestid",
              "x-amzn-trace-id"


       ]
            },
          "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager User Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack
 endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "Boolean"
              },
              "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "Boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "String"
              }

        },
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
```

```
                    "ref": "Endpoint"
                  }
               ]
            }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },

                  true
                ]
              }
            ],
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
            "type": "error"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],

  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
            "type": "error"
          },
          {
            "conditions": [],
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },
              "properties": {},
              "headers": {}
            },
```

```
                          "type": "endpoint"
                        }
                   ]
                },
                {
                   "conditions": [
                      {
                          "fn": "isSet",
                          "argv": [

      {
                               "ref": "Region"
                           }
                      ]
                   }
                ],
                "type": "tree",
                "rules": [
                   {
                      "conditions": [
                         {
                             "fn": "aws.partition",
                             "argv": [
                               {
                                   "ref": "Region"
                               }
                             ],
                             "assign": "PartitionResult"
                         }
                      ],
                      "type": "tree",

     "rules": [
                         {
                             "conditions": [
                               {
                                   "fn": "booleanEquals",
                                   "argv": [
                                     {
                                         "ref": "UseFIPS"
                                     },
                                     true
                                   ]
                               },
                               {
                                   "fn": "booleanEquals",
                                   "argv": [
                                       {
```

```
                                "ref": "UseDualStack"

                              },
                              true
                            ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
                          {
                            "conditions": [
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  true,
                                  {
                                    "fn": "getAttr",

          "argv": [

                                      {
                                        "ref": "PartitionResult"
                                      },
                                      "supportsFIPS"
                                    ]
                                  }
                                ]
                              },
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  true,
                                  {

                  "fn": "getAttr",

                                    "argv": [
                                      {
                                        "ref": "PartitionResult"
                                      },
                                      "supportsDualStack"
                                    ]
                                  }
                                ]
                              }
                            ],
                            "type": "tree",
                            "rules": [
                              {
```

```
        "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ]
                },
                {
                    "conditions": [],
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                        "type": "error"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",

    "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
```

```
                    ]
                  }
                ]
              }
            ],
            "type": "tree",
            "rules": [
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ]
          },
          {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"
          }
        ]
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  true,
                  {
```

```
                                "fn": "getAttr",
                                  "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                  ]
                                }
                              ]
                            }
                          ],
                          "type": "tree",
                          "rules": [
                            {

                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                          ]
                        },
                        {
                          "conditions": [],
                          "error": "DualStack is enabled but this partition does
 not support DualStack",
                          "type": "error"
                        }
                      ]
                    },
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                    }
                  ]
                }
              ]
```

          },
          {
            "conditions":
[],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
          }
        ]
      },
      "smithy.rules#endpointTests": {
        "testCases": [
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
              }
            },

        "params": {
                "Region": "us-east-1",

```
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },

    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
```

```
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
          }
        },

  "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
```

```
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                         "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                         "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
                      }
                    },

    "params": {
                      "Region": "us-gov-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                    },
                    "params": {
                      "Region": "us-iso-east-1",
                      "UseFIPS": true,
                      "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                    "expect":
{
                      "endpoint": {
```

```
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack":
true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
```

```
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
          }
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }

},
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
          }
        },
         "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
          "endpoint": {
            "url": "https://example.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false,
          "Endpoint": "https://example.com"
```

```
                }
            },
            {

"documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "Missing region",
            "expect": {
                "error": "Invalid Configuration: Missing Region"
            }
        }
    ],
    "version": "1.0"
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated":
{
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "IdentityProviderSummaries"
            }
```

```
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
          "type": "structure",
          "members": {
            "IdentityProviderSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
                "smithy.api#required": {}
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
          },
          "errors": [
            {
```

```
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
     },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
```

```
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
            "type":
"structure",
            "members": {
                "InstanceSummaries": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                    "traits": {
                        "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
```

```
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {

  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n      more sets of key-value pairs that you specify.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
              }
            },

      "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
          },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceUserSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {

          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }

          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for a user.</p>",
              "smithy.api#required": {}
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          },
          "SubscriptionStartDate": {
```

```
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
                    }
                },
                "SubscriptionEndDate": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                },
                {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/RegisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n      subnets to provision VPC endpoints.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 404
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",

    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Settings": {
    "type": "structure",
```

```
"members": {
    "Subnets": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
            "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
            "smithy.api#length": {
                "min": 1
            },
            "smithy.api#required": {}
        }
    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n      your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n      subnets to provision VPC endpoints, and the security group ID that
 is associated with the VPC\n      endpoints. The security group should permit inbound TCP port 1688
communication from resources\n      in the VPC.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related costs
will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
 For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n        monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/StartProductSubscription"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
```

```
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
```

```
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/StopProductSubscription"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required":
{}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
```

```
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation":
  "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StringList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {

              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
  request.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
```

```
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity\n     provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
```

```
        "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n     update any combination of settings in a single operation such as the:</p>\n
<ul>\n        <li>\n              <p>Subnets which you want to add to provision VPC endpoints.</p>\n        </li>\n
    <li>\n           <p>Subnets which you want to remove the VPC endpoints from.</p>\n        </li>\n
<li>\n              <p>Security group ID which permits traffic to the VPC endpoints.</p>\n        </li>\n     </ul>",
                "smithy.api#required": {}
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n     require connectivity to activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID
```

that allows inbound TCP port 1688 communication between resources in\n   your VPC and the VPC endpoints for activation servers.</p>"
                  }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as\n   the subnets to provision VPC endpoints.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>A parameter is not valid.</p>",
                "smithy.api#error": "client"
            }
        }
    }
}
{
  "smithy": "2.0",
  "metadata": {
    "suppressions": [
      {
        "id": "HttpMethodSemantics",
        "namespace": "*"
      },
      {
        "id": "HttpResponseCodeSemantics",
        "namespace": "*"
      },
      {
        "id": "PaginatedTrait",
        "namespace": "*"
      },
      {
        "id": "HttpHeaderTrait",
        "namespace": "*"
      },
      {
        "id": "HttpUriConflict",
        "namespace": "*"
      },

```
        {
            "id": "Service",
            "namespace": "*"
        }
    ]
},
"shapes": {
    "com.amazonaws.licensemanager#AWSLicenseManager": {
        "type": "service",
        "version": "2018-08-01",
        "operations": [
            {
                "target": "com.amazonaws.licensemanager#AcceptGrant"
            },

{

                "target": "com.amazonaws.licensemanager#CheckInLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CheckoutLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateGrant"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateGrantVersion"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicense"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
             },
            {
                "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
            },
            {
                "target": "com.amazonaws.licensemanager#CreateToken"
            },
```

```
    {
      "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
      "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    }

},
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
    {
      "target": "com.amazonaws.licensemanager#GetGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseUsage"
    },
    {
      "target": "com.amazonaws.licensemanager#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#ListDistributedGrants"
```

```
    },
    {
      "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
    },
    {

      "target": "com.amazonaws.licensemanager#ListLicenseVersions"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListResourceInventory"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTokens"
    },
    {
      "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
    },
    {
```

```
        "target": "com.amazonaws.licensemanager#RejectGrant"
      },
      {
        "target": "com.amazonaws.licensemanager#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UntagResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager",
        "arnNamespace": "license-manager",
        "cloudFormationName": "LicenseManager",
        "cloudTrailEventSource": "licensemanager.amazonaws.com",
        "endpointPrefix": "license-manager"
      },
      "aws.auth#sigv4": {
        "name": "license-manager"
      },
      "aws.protocols#awsJson1_1": {},
      "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n      Amazon Web Services accounts and on-premises servers.</p>",
      "smithy.api#title": "AWS License Manager",
      "smithy.api#xmlNamespace": {
        "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
      },
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
          "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
```

```
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "Boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
    "type": "Boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "String"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]

            }
        ],
        "type": "tree",
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
```

```
              ]
            }
          ],
          "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
          "type": "error"
        },
        {

"conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ],
          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
          "type": "error"
        },
        {
          "conditions": [],
          "endpoint": {
            "url": {
              "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "type": "tree",
      "rules": [
```

```
                        {
                          "conditions": [
                            {
                               "fn": "aws.partition",
                              "argv": [
                                {
                                   "ref": "Region"
                                }
                              ],
                              "assign": "PartitionResult"
                            }
                          ],
                          "type": "tree",
                          "rules": [
                            {
                              "conditions": [
                                {
                                  "fn": "booleanEquals",
                                  "argv": [
                                    {
                                       "ref":
"UseFIPS"
                                    },
                                     true
                                  ]
                                },
                                {
                                  "fn": "booleanEquals",
                                  "argv": [
                                    {
                                       "ref": "UseDualStack"
                                    },
                                     true
                                  ]
                                }
                              ],
                              "type": "tree",
                              "rules": [
                                {
                                    "conditions": [
                                      {
                                        "fn": "booleanEquals",
                                        "argv": [
                                           true,
                                          {
                                             "fn": "getAttr",
                                              "argv": [
                                                {
```

```json
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        }
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]
                        }
                      ]
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ]
                },
                {
                  "conditions": [],
                  "error": "FIPS and DualStack are enabled, but this partition does not support one or both",
                  "type": "error"
                }
              ]
            },
```

```
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        },
```

```
                                                    {
                                                        "conditions": [],
                                                        "error": "FIPS is enabled but this
partition does not support FIPS",
                                                        "type": "error"
                                                    }
                                                ]
                                            },
                                            {
                                                "conditions": [
                                                    {
                                                        "fn": "booleanEquals",
                                                        "argv": [
                                                            {
                                                                "ref": "UseDualStack"
                                                            },
                                                            true
                                                        ]
                                                    }
                                                ],
                                                "type": "tree",
                                                "rules": [
                                                    {
                                                        "conditions": [
                                                            {
                                                                "fn": "booleanEquals",
                                                                "argv": [
                                                                    true,
                                                                    {
                                                                        "fn": "getAttr",
                                                                        "argv": [
                                                                            {
                                                                                "ref": "PartitionResult"
                                                                            },

"supportsDualStack"
                                                                        ]
                                                                    }
                                                                ]
                                                            }
                                                        ],
                                                        "type": "tree",
                                                        "rules": [
                                                            {
                                                                "conditions": [],
                                                                "endpoint": {
                                                                    "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
```

```
                              "properties": {},
                              "headers": {}

                      },
                            "type": "endpoint"
                        }
                    ]
                },
                {
                    "conditions": [],
                    "error": "DualStack is enabled but this partition does not support DualStack",
                    "type": "error"
                }
            ]
        },
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ]
}
    ]
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint":

{

                    "url": "https://license-manager.af-south-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "af-south-1",
                "UseFIPS": false,
```

```
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "ap-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "ap-northeast-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-2.amazonaws.com"
        }
      },
      "params": {
        "Region": "ap-northeast-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-3.amazonaws.com"
        }
```

```
            },
            "params": {
              "Region": "ap-northeast-3",
              "UseFIPS": false,
              "UseDualStack": false
            }


    },
        {
          "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-1.amazonaws.com"


}
          },
          "params": {
            "Region": "ap-southeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
```

```
      {
        "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-southeast-3.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-southeast-3",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ca-central-1.amazonaws.com"
          }
        },

        "params": {
          "Region": "ca-central-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.eu-central-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "eu-central-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.eu-north-1.amazonaws.com"
          }
```

```
      },
      "params": {
        "Region": "eu-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.eu-south-1.amazonaws.com"
        }
      },
      "params": {
        "Region":
"eu-south-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.eu-west-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "eu-west-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
      "expect": {

        "endpoint": {
          "url": "https://license-manager.eu-west-2.amazonaws.com"
        }
      },
      "params": {
        "Region": "eu-west-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
```

```
{
  "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.eu-west-3.amazonaws.com"
    }
  },
  "params": {
    "Region": "eu-west-3",
    "UseFIPS": false,

    "UseDualStack": false
  }
},
{
  "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.me-south-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "me-south-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.sa-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "sa-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager.us-east-1.amazonaws.com"
    }
  },
  "params": {
```

          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }

  },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-east-2.amazonaws.com"
              }

    },
      "params": {
        "Region": "us-east-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
          {
            "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-east-2.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-2",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region

us-west-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-west-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-west-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-west-1.amazonaws.com"
                    }
                },
                "params": {

        "Region": "us-west-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-west-2.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-west-2",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

"expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-west-2.amazonaws.com"
                    }
                },
                "params": {

```
          "Region": "us-west-2",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,

          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
```

```
        "expect": {
          "endpoint": {
            "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-northwest-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
```

```
                    "UseDualStack": true
                }
            },

        {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-gov-west-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-west-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
```

```
              "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-west-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-gov-east-1.api.aws"
            }
          },

  "params": {
              "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
```

```
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": true,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
                        }
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                        "error": "DualStack is enabled
but this partition does not support DualStack"
                    },
                    "params": {
                        "Region": "us-iso-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
                        }
                    },
                    "params": {
                        "Region": "us-isob-east-1",
                        "UseFIPS": false,
```

```
              "UseDualStack": false
            }
          },

        {
            "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },

      "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

  "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
            },
```

```
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,

    "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
        },
        {
```

```
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanager#AcceptGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

},

    "output": {
        "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"

}
    ],
    "traits": {
        "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
    }
},
"com.amazonaws.licensemanager#AcceptGrantRequest": {
```

```
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#AcceptGrantResponse": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Grant
ARN.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanager#GrantStatus",
              "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#AccessDeniedException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
```

```
      "traits": {
         "aws.protocols#awsQueryError": {

"code": "ServiceAccessDenied",
            "httpResponseCode": 401
         },
         "smithy.api#documentation": "<p>Access to resource denied.</p>",
         "smithy.api#error": "client",
         "smithy.api#httpError": 401
      }
   },
   "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
      "type": "enum",
      "members": {
         "DISTRIBUTED_GRANTS_ONLY": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
            }
         },
         "ALL_GRANTS_PERMITTED_BY_ISSUER": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
            }
         }
      }
   },
   "com.amazonaws.licensemanager#AllowedOperation": {

"type": "enum",
      "members": {
         "CREATE_GRANT": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "CreateGrant"
            }
         },
         "CHECKOUT_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "CheckoutLicense"
            }
         },
         "CHECKOUT_BORROW_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "CheckoutBorrowLicense"
```

```
          }
        },
        "CHECK_IN_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckInLicense"
          }
        },
        "EXTEND_CONSUMPTION_LICENSE":
{

          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ExtendConsumptionLicense"
          }
        },
        "LIST_PURCHASED_LICENSES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ListPurchasedLicenses"
          }
        },
        "CREATE_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateToken"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#AllowedOperationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#AllowedOperation"
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max":
7
        }
      }
    },
    "com.amazonaws.licensemanager#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 2048
```

```
        },
        "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-
z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
      }
    },
    "com.amazonaws.licensemanager#ArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
      }
    },
    "com.amazonaws.licensemanager#AuthorizationException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError":
 {
          "code": "AuthorizationFailure",
          "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n        policy associated with this account.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 403
      }
    },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
      "type": "structure",
      "members": {
        "LastRunTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes automated discovery.</p>"
      }
    },

  "com.amazonaws.licensemanager#Boolean": {
      "type": "boolean",
      "traits": {
```

```
        "smithy.api#default": false
      }
    },
    "com.amazonaws.licensemanager#BorrowConfiguration": {
      "type": "structure",
      "members": {
        "AllowEarlyCheckIn": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxTimeToLiveInMinutes": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
            "smithy.api#required": {}
          }
        }

      },
        "traits": {
          "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#BoxBoolean": {
      "type": "boolean"
    },
    "com.amazonaws.licensemanager#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanager#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#CheckInLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
```

```
                    "target": "com.amazonaws.licensemanager#AuthorizationException"

                },
                {
                    "target": "com.amazonaws.licensemanager#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
            }
        },
        "com.amazonaws.licensemanager#CheckInLicenseRequest":
{
            "type": "structure",
            "members": {
                "LicenseConsumptionToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License consumption token.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Beneficiary": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License beneficiary.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
```

        },
      "com.amazonaws.licensemanager#CheckInLicenseResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicense":

{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
          },
          {
            "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"

```
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
      "type": "structure",
      "members":
{
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
            "smithy.api#required": {}
          }
        },
        "DigitalSignatureMethod": {
          "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
          "traits": {
            "smithy.api#documentation": "<p>Digital signature method. The possible
 value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
            "smithy.api#required": {}
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
```

```
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>"
            }
          },
          "EntitlementsAllowed":
{
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          },
          "SignedToken": {
            "target": "com.amazonaws.licensemanager#SignedToken",
            "traits": {
              "smithy.api#documentation": "<p>Signed token.</p>"
            }
          },
```

```json
        "IssuedAt": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
          }
        },
        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
```

```json
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target":
"com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n          <p>If the
account that created the license is the same that is performing the check out, you must\n          specify the account
as the beneficiary.</p>\n        </note>"
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
        "type": "structure",
        "members": {
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product
SKU.</p>",
              "smithy.api#required": {}
            }
          },
          "CheckoutType": {
            "target": "com.amazonaws.licensemanager#CheckoutType",
            "traits": {
              "smithy.api#documentation": "<p>Checkout type.</p>",
              "smithy.api#required": {}
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
              "smithy.api#required": {}
            }
          },
```

```
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>",

  "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          }
        },
        "traits": {

  "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
        "type": "structure",
        "members": {
          "CheckoutType": {
            "target": "com.amazonaws.licensemanager#CheckoutType",
            "traits": {
              "smithy.api#documentation": "<p>Checkout type.</p>"
            }
          },
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>"
            }
```

```
        },
        "EntitlementsAllowed": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
          }
        },
         "SignedToken": {
          "target": "com.amazonaws.licensemanager#SignedToken",
          "traits": {
            "smithy.api#documentation": "<p>Signed token.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "IssuedAt": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
          }
        },
        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutType": {
      "type": "enum",
      "members": {
        "PROVISIONAL": {
          "target": "smithy.api#Unit",
```

```
          "traits": {
            "smithy.api#enumValue": "PROVISIONAL"
          }
        },
        "PERPETUAL": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PERPETUAL"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ClientRequestToken": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 36
        }
      }
    },
    "com.amazonaws.licensemanager#ClientToken": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 2048
        },
        "smithy.api#pattern": "^\\S+$"
      }
    },
    "com.amazonaws.licensemanager#ConflictException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "ConflictException",
          "httpResponseCode":
409
        },
        "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 409
```

```
            }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
            "type": "structure",
            "members": {
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#ResourceType",
                    "traits": {
                        "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
                    }
                },
                "ConsumedLicenses": {
                    "target": "com.amazonaws.licensemanager#BoxLong",
                    "traits": {
                        "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
                    }
                }
            },

            "traits": {
                "smithy.api#documentation": "<p>Details about license consumption.</p>"
            }
        },
        "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
            }
        },
        "com.amazonaws.licensemanager#ConsumptionConfiguration": {
            "type": "structure",
            "members": {
                "RenewType": {
                    "target": "com.amazonaws.licensemanager#RenewType",
                    "traits": {
                        "smithy.api#documentation": "<p>Renewal frequency.</p>"
                    }
                },
                "ProvisionalConfiguration": {
                    "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
                    }
                },
                "BorrowConfiguration": {
                    "target": "com.amazonaws.licensemanager#BorrowConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
```

```
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target":
"com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n       entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrantRequest": {
```

```
        "type": "structure",
        "members": {
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant name.</p>",
                    "smithy.api#required": {}

    }
        },
        "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                "smithy.api#required": {}
            }
        },
        "Principals": {
            "target": "com.amazonaws.licensemanager#PrincipalArnList",
            "traits": {
                "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n        (ARN):</p>\n        <ul>\n            <li>\n                <p>An Amazon Web Services
account, which includes only the account specified.</p>\n            </li>\n        </ul>\n        <ul>\n            <li>\n
    <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n            </li>\n        </ul>\n
<ul>\n
        <li>\n                <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n        </ul>",
                "smithy.api#required": {}
            }
        },
        "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                "smithy.api#required": {}
            }
        },
        "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
```

```
              "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },

      "com.amazonaws.licensemanager#CreateGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateGrantVersion":
    {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
      "type": "structure",
      "members": {
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

  "smithy.api#required": {}
          }
        },
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
              "smithy.api#documentation": "<p>Grant name.</p>"
            }
          },
          "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>Grant status reason.</p>"
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the grant.</p>"
            }
          },
          "Options": {
            "target": "com.amazonaws.licensemanager#Options",
            "traits": {
              "smithy.api#documentation": "<p>The options specified for the grant.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    }
  },
  "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Grant ARN.</p>"
        }
      },
      "Status": {
```

```
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>New version of the grant.</p>"
                }
            }
        },
        "traits":
{
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicense": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
```

```
          "traits": {
            "smithy.api#documentation": "<p>Creates a license.</p>"
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
 all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of
licenses purchased and used.</p>"
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Name of the license configuration.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the license configuration.</p>"
            }
          },
          "LicenseCountingType": {
            "target": "com.amazonaws.licensemanager#LicenseCountingType",
            "traits": {
              "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
            }
          },
          "LicenseCountHardLimit": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n        blocks the launch of new instances.</p>"
            }
          },
          "LicenseRules": {
            "target":
 "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n        vary by dimension, as follows.</p>\n
<ul>\n        <li>\n          <p>\n            <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
        <code>licenseAffinityToHost</code> |\n            <code>maximumCores</code> |
<code>minimumCores</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n            <code>maximumCores</code> |
<code>minimumCores</code> |\n            <code>maximumSockets</code> | <code>minimumSockets</code> |\n
        <code>maximumVcpus</code> | <code>minimumVcpus</code>\n          </p>\n        </li>\n
<li>\n
  <p>\n            <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n            <code>maximumSockets</code> |
<code>minimumSockets</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n            <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n          </p>\n        </li>\n      </ul>\n        <p>The unit for
```

<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for <code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n <code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n <code>True</code> and <code>False</code>.</p>"

```
                }
            },

    "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                    "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
                }
            },
            "DisassociateWhenNotFound": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            },
            "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                    "smithy.api#documentation": "<p>Product information.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
        "type": "operation",
```

```
"input": {
  "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
},
"output": {
  "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanager#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanager#AuthorizationException"
  },
  {
    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
  },
  {
    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
  },
  {
    "target": "com.amazonaws.licensemanager#ServerInternalException"
  },
  {
    "target": "com.amazonaws.licensemanager#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
}
},
"com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {

      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
        "smithy.api#required": {}
      }
    },
    "SourceLicenseContext": {
      "target": "com.amazonaws.licensemanager#LicenseConversionContext",
      "traits": {
        "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n      For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
```

```
the <i>License Manager User Guide</i>.</p>",
                    "smithy.api#required": {}
                }
            },
            "DestinationLicenseContext":
{
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

                "traits": {
                    "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
```

```
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "ReportGeneratorName":
{
          "target": "com.amazonaws.licensemanager#ReportGeneratorName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>",
            "smithy.api#required": {}
          }
        },
        "Type": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n        <ul>\n          <li>\n              <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n          </li>\n          <li>\n              <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n          </li>\n
</ul>",
            "smithy.api#required": {}

        }
        },
```

```
      "ReportContext": {
        "target": "com.amazonaws.licensemanager#ReportContext",
        "traits": {
          "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
          "smithy.api#required": {}
        }
      },
      "ReportFrequency": {
        "target": "com.amazonaws.licensemanager#ReportFrequency",
        "traits": {
          "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
          "smithy.api#required": {}
        }
      },
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientRequestToken",
        "traits": {
          "smithy.api#documentation": "<p>Unique,
 case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
          "smithy.api#required": {}
        }
      },
      "Description": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Description of the report generator.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanager#TagList",
        "traits": {
          "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
    "type": "structure",
    "members": {
      "LicenseManagerReportGeneratorArn":
{
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
```

```
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product
name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product SKU.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits": {
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region for the license.</p>",
              "smithy.api#required": {}
```

```
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
          "smithy.api#required": {}
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>",
          "smithy.api#required": {}
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License beneficiary.</p>",
          "smithy.api#required": {}
        }
      },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n          running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n          usage.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
          "smithy.api#documentation": "<p>Information about the license.</p>"
        }
      },
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientToken",
        "traits": {
          "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
          "smithy.api#required": {}
        }
      }
    },
```

```
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersion": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
    "type": "structure",
    "members": {
      "LicenseArn": {
        "target":
"com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License name.</p>",
          "smithy.api#required": {}
        }
      },
      "ProductName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product name.</p>",
          "smithy.api#required": {}
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#Issuer",
```

```
                    "traits":
{
                        "smithy.api#documentation": "<p>License issuer.</p>",
                        "smithy.api#required": {}
                    }
                },
                "HomeRegion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Home Region of the license.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Validity": {
                    "target": "com.amazonaws.licensemanager#DatetimeRange",
                    "traits": {
                        "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
                        "smithy.api#required": {}
                    }
                },
                "LicenseMetadata": {
                    "target": "com.amazonaws.licensemanager#MetadataList",
                    "traits": {
                        "smithy.api#documentation": "<p>Information about the license.</p>"
                    }
                },
                "Entitlements": {
                    "target": "com.amazonaws.licensemanager#EntitlementList",
                    "traits": {
                        "smithy.api#documentation": "<p>License entitlements.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ConsumptionConfiguration": {
                    "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n          running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n          usage.</p>",
                        "smithy.api#required": {}
                    }
                },

    "Status": {
                    "target": "com.amazonaws.licensemanager#LicenseStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>License status.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the license.</p>"
            }
          }
        },
        "traits": {

  "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New version of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
```

```
{}
        }
    },
    "com.amazonaws.licensemanager#CreateToken": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateTokenRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateTokenResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },

            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateTokenRequest": {
        "type": "structure",
        "members": {
            "LicenseArn": {
```

```
          "target": "com.amazonaws.licensemanager#Arn",
          "traits":
{
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
              "smithy.api#required": {}
          }
        },
        "RoleArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n        License Manager does not check whether the roles are in use.</p>"
          }
        },
        "ExpirationInDays": {
          "target": "com.amazonaws.licensemanager#Integer",
          "traits": {
            "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
          }
        },
        "TokenProperties": {
           "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
            "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n        to the amr claim of the JWT token.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateTokenResponse": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
```

```
          "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>"
            }
          },
          "TokenType": {
            "target": "com.amazonaws.licensemanager#TokenType",
            "traits": {
              "smithy.api#documentation": "<p>Token type.</p>"
            }
          },
          "Token": {
            "target": "com.amazonaws.licensemanager#TokenString",
            "traits": {
              "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DateTime": {
      "type": "timestamp"
    },
    "com.amazonaws.licensemanager#DatetimeRange": {
      "type": "structure",
      "members": {
        "Begin": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Start of the time range.</p>",
            "smithy.api#required": {}
          }
        },
        "End": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>End of the time range.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteGrant": {
      "type": "operation",
      "input": {
```

```
              "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
            },
          "output": {
              "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
          }
        },
        "com.amazonaws.licensemanager#DeleteGrantRequest": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
              }
            },
            "StatusReason": {
              "target": "com.amazonaws.licensemanager#StatusReasonMessage",
              "traits": {
                "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
              }
```

```
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the
grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation":
"<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
```

          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deletes the specified license.</p>"
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },

```
              {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n          <p>You cannot
delete a license configuration that is in use.</p>"
            }
          },
          "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
            "type": "structure",
            "members":
{
              "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>ID of the license configuration.</p>",
                  "smithy.api#required": {}
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
            "type": "structure",
            "members": {},
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
            },
            "errors": [
              {
```

```
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {

  "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
```

```
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "SourceVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the license.</p>",
            "smithy.api#required":
{}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseResponse": {
      "type": "structure",
      "members": {
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "DeletionDate": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
```

```
    },
    "com.amazonaws.licensemanager#DeleteToken": {
      "type": "operation",
       "input": {
        "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },

    {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenRequest": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>",
            "smithy.api#required": {}
          }
        }
```

```
      },
      "traits": {
        "smithy.api#input": { }
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenResponse": {
      "type": "structure",
      "members": { },
      "traits": {
        "smithy.api#output": { }


    }
    },
    "com.amazonaws.licensemanager#DigitalSignatureMethod": {
      "type": "enum",
      "members": {
        "JWT_PS384": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "JWT_PS384"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#Entitlement": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement name.</p>",
            "smithy.api#required": { }
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#Long",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
          }
        },
        "Overage": {
```

```
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement unit.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowCheckIn": {
          "target":
"com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementData": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement data name.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement
data value.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement data unit.</p>",
            "smithy.api#required": {}
          }
        }
```

```
        },
      "traits": {
        "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementDataList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementData"
      }
    },
    "com.amazonaws.licensemanager#EntitlementDataUnit": {
      "type": "enum",
      "members": {
        "COUNT": {
          "target": "smithy.api#Unit",
          "traits": {

"smithy.api#enumValue": "Count"
          }
        },
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "SECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Seconds"
          }
        },
        "MICROSECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Microseconds"
          }
        },
        "MILLISECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Milliseconds"
          }
        },
        "BYTES": {
          "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS":
{
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits"
      }
    },
    "KILOBITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobits"
      }
    },
    "MEGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits"
      }
    },
    "GIGABITS": {
```

```
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits"
        }
      },
      "TERABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits"

}
      },
      "PERCENT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Percent"
        }
      },
      "BYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes/Second"
        }
      },
      "KILOBYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes/Second"
        }
      },
      "MEGABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes/Second"
        }
      },
      "GIGABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
         "traits": {
          "smithy.api#enumValue": "Gigabytes/Second"
        }
      },
      "TERABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabytes/Second"
        }
      },
```

```
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
          }
        },
        "TERABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits/Second"
          }
        },
        "COUNT_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count/Second"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Entitlement"
      }
    },
    "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
      "type": "structure",
```

```
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#EntitlementUnit": {
      "type": "enum",
      "members": {
        "COUNT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count"
          }
        },
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "SECONDS":
{
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Seconds"
          }
        },
        "MICROSECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Microseconds"
          }
        },
        "MILLISECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Milliseconds"
          }
        },
        "BYTES": {
          "target": "smithy.api#Unit",
          "traits": {
```

```json
      "smithy.api#enumValue": "Bytes"
    }
  },
  "KILOBYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Kilobytes"
    }
  },
  "MEGABYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Megabytes"
    }
  },
  "GIGABYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Gigabytes"
    }
  },
  "TERABYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Terabytes"
    }
  },
  "BITS": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Bits"
    }
  },
  "KILOBITS": {
    "target": "smithy.api#Unit",
    "traits": {

      "smithy.api#enumValue": "Kilobits"
    }
  },
  "MEGABITS": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Megabits"
    }
  },
  "GIGABITS": {
    "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue": "Gigabits"
      }
    },
    "TERABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabits"
      }
    },
    "PERCENT": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Percent"
      }
    },
    "BYTES_PER_SECOND": {
      "target":
"smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes/Second"
      }
    },
    "KILOBYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes/Second"
      }
    },
    "MEGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes/Second"
      }
    },
    "GIGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes/Second"
      }
    },
    "TERABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {

"smithy.api#enumValue": "Terabytes/Second"
      }
    },
```

```
    "BITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits/Second"
      }
    },
    "KILOBITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobits/Second"
      }
    },
    "MEGABITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits/Second"
      }
    },
    "GIGABITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabits/Second"
      }
    },

  "TERABITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabits/Second"
      }
    },
    "COUNT_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Count/Second"
      }
    }
  }
},
"com.amazonaws.licensemanager#EntitlementUsage": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Entitlement usage name.</p>",
        "smithy.api#required": {}
      }
```

```
          },
          "ConsumedValue": {
            "target": "com.amazonaws.licensemanager#String",

          "traits": {
              "smithy.api#documentation": "<p>Resource usage consumed.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxCount": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#EntitlementUsageList":
{
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#EntitlementUsage"
          }
      },
      "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
        }
      },
      "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
        "type": "structure",
        "members": {
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>",
              "smithy.api#required": {}
            }
          },
          "DryRun": {
            "target": "com.amazonaws.licensemanager#Boolean",
            "traits": {
              "smithy.api#default": false,
              "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
        "type": "structure",
        "members": {
```

```
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },


        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#FailedDependencyException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        },
        "ErrorCode": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "FailedDependency",
          "httpResponseCode": 424
        },
        "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 424
      }
    },
    "com.amazonaws.licensemanager#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#FilterName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
          }
        },
```

```
      "Values": {
        "target": "com.amazonaws.licensemanager#FilterValues",
        "traits": {
          "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n       describe operation. Filters can be used
to match a set of resources by specific criteria,\n       such as tags, attributes, or IDs.</p>"
    }
  },
  "com.amazonaws.licensemanager#FilterLimitExceededException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "FilterLimitExceeded",
        "httpResponseCode": 400
      },
      "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Filter"
    }
  },
  "com.amazonaws.licensemanager#FilterName": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValue": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValues": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#FilterValue",
      "traits": {
```

```
            "smithy.api#xmlName": "item"
          }
        }
      },
      "com.amazonaws.licensemanager#Filters": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Filter",
          "traits": {
            "smithy.api#xmlName": "item"
          }
        }
      },
      "com.amazonaws.licensemanager#GetAccessToken": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
tokens\n        are valid for one hour.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetAccessTokenRequest": {
        "type": "structure",
        "members": {
          "Token": {
```

```
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
            "smithy.api#required": {}
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
            "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
      "type": "structure",
      "members": {
        "AccessToken": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Temporary access token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetGrantRequest": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrantResponse": {
      "type": "structure",
      "members": {
        "Grant": {
          "target": "com.amazonaws.licensemanager#Grant",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>Grant details.</p>"
                }
            }
        },
        "traits": {

"smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicense": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"

        },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#GetLicenseConfiguration": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
        },
        "output": {
```

```
              "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
              {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
            "type": "structure",
            "members": {
              "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                  "smithy.api#required": {}
                }
              }
            },
            "traits":
{
              "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
            "type": "structure",
            "members": {
              "LicenseConfigurationId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
```

```
        }
      },
      "LicenseConfigurationArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
        }
      },
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Name of the license configuration.</p>"
        }
      },
      "Description": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Description of the license configuration.</p>"
        }
      },
      "LicenseCountingType": {
        "target": "com.amazonaws.licensemanager#LicenseCountingType",
        "traits": {
          "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",

        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
```

```
              "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
          }
      },
      "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
              "smithy.api#documentation": "<p>License configuration status.</p>"
          }

      },
          "OwnerAccountId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                  "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
              }
          },
          "ConsumedLicenseSummaryList": {
              "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
              "traits": {
                  "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
              }
          },
          "ManagedResourceSummaryList": {
              "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
              "traits": {
                  "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
              }
          },
          "Tags": {
              "target": "com.amazonaws.licensemanager#TagList",
               "traits": {
                  "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
              }
          },
          "ProductInformationList": {
              "target": "com.amazonaws.licensemanager#ProductInformationList",
              "traits": {
                  "smithy.api#documentation": "<p>Product information.</p>"
              }
          },
          "AutomatedDiscoveryInformation": {
              "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
              "traits": {
                  "smithy.api#documentation": "<p>Automated discovery information.</p>"
              }
          },
          "DisassociateWhenNotFound": {
              "target": "com.amazonaws.licensemanager#BoxBoolean",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTask": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {

"target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
        }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                "traits": {
                    "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
```

```
                    "smithy.api#required": {}
                  }
                }
              },
            "traits": {
              "smithy.api#input": {}
            }
          },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
            }
```

```
            },

    "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                }
            },
            "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
            "type": "structure",
            "members": {
                "LicenseManagerReportGeneratorArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
                        "smithy.api#required":
{}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
            "type": "structure",
            "members": {
                "ReportGenerator": {
                    "target": "com.amazonaws.licensemanager#ReportGenerator",
                    "traits": {
                        "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
```

```
        },
        "com.amazonaws.licensemanager#GetLicenseRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License version.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseResponse": {
            "type": "structure",
            "members": {
                "License": {
                    "target": "com.amazonaws.licensemanager#License",
                    "traits": {
                        "smithy.api#documentation": "<p>License details.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },

        "com.amazonaws.licensemanager#GetLicenseUsage": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
            },
            "errors": [
                {
```

```
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {

    "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
        "type": "structure",
        "members": {
          "LicenseUsage": {
            "target":
"com.amazonaws.licensemanager#LicenseUsage",
            "traits": {
              "smithy.api#documentation": "<p>License usage details.</p>"
            }
          }
```

```
          },
          "traits": {
             "smithy.api#output": {}
          }
      },
      "com.amazonaws.licensemanager#GetServiceSettings": {
          "type": "operation",
          "input": {
             "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
          },
          "output": {
             "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
          },
          "errors": [
             {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
             },
             {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
             },
             {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
             },
             {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
             }
          ],
          "traits": {
             "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
          }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
          "type": "structure",
          "members": {},
          "traits": {
             "smithy.api#input": {}
          }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
             "S3BucketArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                   "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"
```

```
        }
      },
      "SnsTopicArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
        }
      },
      "OrganizationConfiguration": {
        "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n        cross-account discovery.</p>"
        }
      },
      "EnableCrossAccountsDiscovery": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
        }

      },
      "LicenseManagerResourceShareArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n        provides member accounts with access to this share.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#Grant": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantName": {
```

```
  "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant name.</p>",
              "smithy.api#required": {}
            }
          },
          "ParentArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Parent ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "GranteePrincipalArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {

"smithy.api#documentation": "<p>The grantee principal ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region of the grant.</p>",
              "smithy.api#required": {}
            }
          },
          "GrantStatus": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>",
              "smithy.api#required": {}
            }
          },
          "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>Grant status reason.</p>"
            }
          },
```

```
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant version.</p>",
          "smithy.api#required": {}
        }
      },
      "GrantedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
        "traits": {
          "smithy.api#documentation": "<p>Granted operations.</p>",
          "smithy.api#required": {}
        }
      },
      "Options": {
        "target": "com.amazonaws.licensemanager#Options",
        "traits": {
          "smithy.api#documentation": "<p>The options specified for the grant.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>Describes a grant.</p>"
    }
  },
  "com.amazonaws.licensemanager#GrantList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Grant"
    }
  },
  "com.amazonaws.licensemanager#GrantStatus": {
    "type": "enum",
    "members": {
      "PENDING_WORKFLOW": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_WORKFLOW"
        }
      },
      "PENDING_ACCEPT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_ACCEPT"
        }
      },
      "REJECTED": {
```

```
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "REJECTED"

          }
        },
        "ACTIVE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ACTIVE"
            }
        },
        "FAILED_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED_WORKFLOW"
            }
        },
        "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DELETED"
            }
        },
        "PENDING_DELETE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_DELETE"
            }
        },
        "DISABLED": {
            "target": "smithy.api#Unit",
            "traits":
{

              "smithy.api#enumValue": "DISABLED"
            }
        },
        "WORKFLOW_COMPLETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "WORKFLOW_COMPLETED"
            }
        }
      }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
      "type": "structure",
      "members": {
```

```
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
        }
      },
      "LicenseName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License name.</p>"
        }

    },
      "ProductName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product name.</p>"
        }
      },
      "ProductSKU": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product SKU.</p>"
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#IssuerDetails",
        "traits": {
          "smithy.api#documentation": "<p>Granted license issuer.</p>"
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

      }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
        "traits": {
          "smithy.api#documentation": "<p>Granted license status.</p>"
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
```

ISO8601-UTC format.</p>"
              }
            },
            "Beneficiary": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
              }
            },
            "Entitlements": {
              "target": "com.amazonaws.licensemanager#EntitlementList",
              "traits": {
                "smithy.api#documentation": "<p>License entitlements.</p>"
              }
            },
            "ConsumptionConfiguration": {
              "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
              "traits": {
                "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
              }
            },
            "LicenseMetadata": {
              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                "smithy.api#documentation": "<p>Granted license metadata.</p>"
              }
            },
            "CreateTime": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

  }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Version of the granted license.</p>"
              }
            },
            "ReceivedMetadata": {
              "target": "com.amazonaws.licensemanager#ReceivedMetadata",
              "traits": {
                "smithy.api#documentation": "<p>Granted license received metadata.</p>"
              }
            }
          },
          "traits": {

```
            "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
          }
        },
        "com.amazonaws.licensemanager#GrantedLicenseList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#GrantedLicense"
          }
        },
        "com.amazonaws.licensemanager#ISO8601DateTime":
  {
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-
  1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
          }
        },
        "com.amazonaws.licensemanager#Integer": {
          "type": "integer"
        },
        "com.amazonaws.licensemanager#InvalidParameterValueException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "InvalidParameterValueProvided",
              "httpResponseCode": 400
            },
            "smithy.api#documentation":
  "<p>One or more parameter values are not valid.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#InvalidResourceStateException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
```

```
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidResourceState",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },

    "com.amazonaws.licensemanager#InventoryFilter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the filter.</p>",
              "smithy.api#required": {}
            }
          },
          "Condition": {
            "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
            "traits": {
              "smithy.api#documentation": "<p>Condition of the filter.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }

      },
        "traits": {
          "smithy.api#documentation": "<p>An inventory filter.</p>"
        }
      },
    "com.amazonaws.licensemanager#InventoryFilterCondition": {
        "type": "enum",
        "members": {
          "EQUALS": {
            "target": "smithy.api#Unit",
```

```
          "traits": {
            "smithy.api#enumValue": "EQUALS"
          }
        },
        "NOT_EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "NOT_EQUALS"
          }
        },
        "BEGINS_WITH": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "BEGINS_WITH"
          }
        },
        "CONTAINS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CONTAINS"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#InventoryFilter"
      }
    },
    "com.amazonaws.licensemanager#Issuer": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Issuer name.</p>",
            "smithy.api#required": {}
          }
        },
        "SignKey": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
          }
        }
```

```
          },
        "traits": {
          "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#IssuerDetails": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>"
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#License": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

          "smithy.api#documentation": "<p>License name.</p>"
            }
```

```
        },
        "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Product name.</p>"
            }
        },
        "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Product SKU.</p>"
            }
        },
        "Issuer": {
            "target": "com.amazonaws.licensemanager#IssuerDetails",
            "traits": {
                "smithy.api#documentation": "<p>License issuer.</p>"
            }
        },
        "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

                "smithy.api#documentation": "<p>Home Region of the license.</p>"
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
                "smithy.api#documentation": "<p>License status.</p>"
            }
        },
        "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
                "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
            }
        },
        "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
        },
        "Entitlements": {
```

```
        "target": "com.amazonaws.licensemanager#EntitlementList",
                "traits": {
                    "smithy.api#documentation": "<p>License entitlements.</p>"
                }
            },
            "ConsumptionConfiguration": {
                "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
                }
            },
            "LicenseMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>License metadata.</p>"
                }
            },
            "CreateTime": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>License
creation time.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License version.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseConfiguration": {
        "type": "structure",
        "members": {
            "LicenseConfigurationId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
                }
            },
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
```

```
        }
      },
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Name of the license configuration.</p>"
        }
      },
      "Description": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Description of the license configuration.</p>"
        }
      },
      "LicenseCountingType": {
        "target": "com.amazonaws.licensemanager#LicenseCountingType",
        "traits": {
          "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
        }
      },
      "DisassociateWhenNotFound": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
```

```
              "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
          }
      },
      "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
              "smithy.api#documentation": "<p>Status of the license configuration.</p>"
          }
      },
      "OwnerAccountId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
              "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

          }
      },
      "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
              "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
          }
      },
      "ManagedResourceSummaryList": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
          "traits": {
              "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
          }
      },
      "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
              "smithy.api#documentation": "<p>Product information.</p>"
          }
      },
      "AutomatedDiscoveryInformation": {

          "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
          "traits": {
              "smithy.api#documentation": "<p>Automated discovery information.</p>"
          }
      }
  },
  "traits": {
      "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n        consumed and enforced by License Manager. Components include specifications for
the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n        must be associated with a
host), and the number of licenses purchased and used.</p>"
```

```
          }
        },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
        "type": "structure",
        "members": {
          "ResourceArn":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of server resource.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
            }
          },
          "AssociationTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
            }
          },
          "AmiAssociationScope": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
```

```
            }
        },
        "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
            "type": "enum",
            "members":
{
                "AVAILABLE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "AVAILABLE"
                    }
                },
                "DISABLED": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "DISABLED"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
            "type": "structure",
            "members": {
                "ResourceArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
                    }
                },
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#ResourceType",
                    "traits":
{
                        "smithy.api#documentation": "<p>Type of resource.</p>"
                    }
                },
                "ResourceStatus": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Status of the resource.</p>"
                    }
                },
                "ResourceOwnerId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
                    }
                },
```

```
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
          }
        },

    "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurations": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfiguration"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
      "members": {
        "UsageOperation": {
          "target": "com.amazonaws.licensemanager#UsageOperation",
          "traits": {
            "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n        </p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {

            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
              }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the conversion task.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "StartTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
```

```
        "traits": {
            "smithy.api#documentation":
"<p>The time the conversion task was started at.</p>"
        }
    },
    "LicenseConversionTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {
            "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
        }
    },
    "EndTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {
            "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
}
},
"com.amazonaws.licensemanager#LicenseConversionTaskId": {
    "type": "string",
    "traits": {

        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
    }
},
"com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
    "type": "enum",
    "members": {
        "IN_PROGRESS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "IN_PROGRESS"
            }
        },
        "SUCCEEDED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "SUCCEEDED"
            }
```

```
          },
          "FAILED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTasks":
{
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTask"
        }
      },
      "com.amazonaws.licensemanager#LicenseCountingType": {
        "type": "enum",
        "members": {
          "VCPU": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "vCPU"
            }
          },
          "INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Instance"
            }
          },
          "CORE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Core"
            }
          },
          "SOCKET": {
            "target": "smithy.api#Unit",
             "traits": {
              "smithy.api#enumValue": "Socket"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseDeletionStatus": {
        "type": "enum",
        "members": {
```

```
      "PENDING_DELETE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_DELETE"
        }
      },
      "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DELETED"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#LicenseList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#License"
    }
  },
  "com.amazonaws.licensemanager#LicenseOperationFailure": {

    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Resource type.</p>"
        }
      },
      "ErrorMessage": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Error message.</p>"
        }
      },
      "FailureTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {

"smithy.api#documentation": "<p>Failure time.</p>"
        }
```

```
          },
          "OperationName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the operation.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
            }
          },
          "OperationRequestedBy": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
            }
          },
          "MetadataList": {

            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Reserved.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseOperationFailureList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecification": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
```

```
      "smithy.api#required": {}
     }
    },
    "AmiAssociationScope": {
     "target": "com.amazonaws.licensemanager#String",
     "traits": {
      "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
     }
    }
   },
   "traits": {
    "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
   }
  },
  "com.amazonaws.licensemanager#LicenseSpecifications": {
   "type": "list",
   "member": {
    "target": "com.amazonaws.licensemanager#LicenseSpecification"
   }
  },
  "com.amazonaws.licensemanager#LicenseStatus": {
   "type": "enum",
   "members": {
    "AVAILABLE": {
     "target": "smithy.api#Unit",

     "traits": {
      "smithy.api#enumValue": "AVAILABLE"
     }
    },
    "PENDING_AVAILABLE": {
     "target": "smithy.api#Unit",
     "traits": {
      "smithy.api#enumValue": "PENDING_AVAILABLE"
     }
    },
    "DEACTIVATED": {
     "target": "smithy.api#Unit",
     "traits": {
      "smithy.api#enumValue": "DEACTIVATED"
     }
    },
    "SUSPENDED": {
     "target": "smithy.api#Unit",
     "traits": {
      "smithy.api#enumValue": "SUSPENDED"
     }
```

```
        },
        "EXPIRED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EXPIRED"
          }
        },

      "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseUsage": {
      "type": "structure",
      "members": {
        "EntitlementUsages": {
          "target": "com.amazonaws.licensemanager#EntitlementUsageList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlement usages.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

      }
    },
    "com.amazonaws.licensemanager#LicenseUsageException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "LicenseUsageFailure",
          "httpResponseCode": 412
```

```
          },
          "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 412
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
        },
        "output": {

    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
        "type": "structure",
        "members": {
            "LicenseConfigurationAssociations": {
                "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
                }
            },
            "NextToken": {

"target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListDistributedGrants": {
        "type": "operation",
        "input": {
```

```
          "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
        },
      "output": {
          "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
        },
      "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
      "traits": {
          "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
        "type": "structure",
        "members": {
          "GrantArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {

          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n    <ul>\n      <li>\n        <p>\n          <code>LicenseArn</code>\n        </p>\n
    </li>\n      <li>\n        <p>\n          <code>GrantStatus</code>\n        </p>\n        </li>\n
```

```
          <li>\n        <p>\n            <code>GranteePrincipalARN</code>\n         </p>\n     </li>\n
<li>\n        <p>\n            <code>ProductSKU</code>\n         </p>\n     </li>\n     <li>\n
<p>\n             <code>LicenseIssuerName</code>\n
     </p>\n     </li>\n     </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
        "type": "structure",
        "members": {
          "Grants": {
            "target": "com.amazonaws.licensemanager#GrantList",

   "traits": {
              "smithy.api#documentation": "<p>Distributed grant details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
        },
        "errors": [
          {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
```

```
        },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
      "type": "structure",
      "members": {
        "LicenseOperationFailureList": {
          "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
          "traits": {
            "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
      },

    "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
```

```
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the license configurations
 for your account.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArns": {
                "target": "com.amazonaws.licensemanager#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanager#Filters",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n           <li>\n              <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n           Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n           Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n           </li>\n           <li>\n
<p>\n              <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n           Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
   </li>\n           <li>\n              <p>\n              <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n           Logical operators are
```

```
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n            </li>\n        </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
        "type": "structure",
        "members": {
            "LicenseConfigurations": {
                "target": "com.amazonaws.licensemanager#LicenseConfigurations",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license configurations.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",

                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
```

```
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
          "type": "structure",
          "members": {
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#Filters",
              "traits": {
                "smithy.api#documentation": "<p>\n        Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n      </p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
          "type": "structure",
          "members": {
            "LicenseConversionTasks": {
              "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
              "traits": {
                "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
              }
            },
```

          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target":
"com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"

```
            }
        },
        "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
            "type": "structure",
            "members": {
                "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n        <ul>\n          <li>\n            <p>\n              <code>LicenseConfigurationArn</code>\n
</p>\n          </li>\n        </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits":
{
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
            "type": "structure",
            "members": {
                "ReportGenerators": {
                    "target": "com.amazonaws.licensemanager#ReportGeneratorList",
                    "traits": {
                        "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
```

```
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
```

```
              "traits": {

        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {
        "LicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
          }
        },
        "NextToken": {
           "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
```

```
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
        "type": "structure",
        "members": {
          "Licenses": {
```

```
                "target": "com.amazonaws.licensemanager#LicenseList",
                "traits": {
                    "smithy.api#documentation": "<p>License details.</p>"
                }
            },

"NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenses": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicensesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListLicensesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
        }
```

        },
        "com.amazonaws.licensemanager#ListLicensesRequest": {
          "type": "structure",
          "members": {
            "LicenseArns": {
              "target": "com.amazonaws.licensemanager#ArnList",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
              }
            },

          "Filters": {
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                  "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>Beneficiary</code>\n   </p>\n
  \n   \n   <p>\n   <code>ProductSKU</code>\n   </p>\n   \n
  \n   <p>\n   <code>Fingerprint</code>\n   </p>\n   \n   \n
  <p>\n   <code>Status</code>\n   </p>\n   \n   "
                }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token
for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicensesResponse": {
          "type": "structure",
          "members": {
            "Licenses": {
              "target": "com.amazonaws.licensemanager#LicenseList",
              "traits": {
                "smithy.api#documentation": "<p>License details.</p>"
              }
            },

```
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {

    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrants": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n        recipient as this Amazon Web Services account, your organization, or an organizational unit\n
    (OU) to which this member account belongs.</p>"
      }
```

```
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
            "smithy.api#required": {}
          }
        },
        "Filters": {
```

```
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
    <ul>\n        <li>\n            <p>\n                <code>ParentArn</code>\n            </p>\n        </li>\n
<li>\n            <p>\n                <code>GranteePrincipalArn</code>\n            </p>\n        </li>\n    </ul>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
        "type": "structure",
        "members": {
            "Grants": {
                "target": "com.amazonaws.licensemanager#GrantList",
                "traits": {
                    "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
        "type": "structure",
```

```
"members": {
  "GrantArns": {
    "target": "com.amazonaws.licensemanager#ArnList",
     "traits": {
       "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
     }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanager#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n      <ul>\n         <li>\n            <p>\n               <code>ProductSKU</code>\n            </p>\n
   </li>\n         <li>\n            <p>\n               <code>LicenseIssuerName</code>\n            </p>\n
</li>\n         <li>\n            <p>\n               <code>LicenseArn</code>\n            </p>\n         </li>\n
<li>\n            <p>\n               <code>GrantStatus</code>\n            </p>\n         </li>\n         <li>\n
<p>\n
      <code>GranterAccountId</code>\n            </p>\n         </li>\n      </ul>"
    }
  },
  "NextToken": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
      "smithy.api#documentation": "<p>Token for the next set of results.</p>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanager#MaxSize100",
    "traits": {
      "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
  "type": "structure",
  "members": {
    "Grants": {
      "target": "com.amazonaws.licensemanager#GrantList",
       "traits": {
         "smithy.api#documentation": "<p>Received grant details.</p>"
       }
    },
    "NextToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
```

```
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#ListReceivedLicenses": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists received licenses.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
  {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
```

```
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {

          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>Beneficiary</code>\n            </p>\n
    </li>\n          <li>\n            <p>\n              <code>ProductSKU</code>\n
        </p>\n          </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
```

```
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#GrantedLicenseList",
          "traits": {
            "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },

        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n          <ul>\n          <li>\n          <p>\n          <code>ProductSKU</code>\n          </p>\n
```

    \n   \n   <p>\n   <code>Status</code>\n   </p>\n   \n
\n   <p>\n   <code>Fingerprint</code>\n   </p>\n   \n   \n
<p>\n   <code>IssuerName</code>\n   </p>\n   \n   \n   <p>\n
<code>Beneficiary</code>\n   </p>\n   \n   "
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
            "type": "structure",
            "members": {
                "Licenses": {
                    "target": "com.amazonaws.licensemanager#GrantedLicenseList",
                    "traits": {
                        "smithy.api#documentation": "<p>Received license details.</p>"
                    }
                },

                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListResourceInventory": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"

```
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#FailedDependencyException"
            },
            {
              "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#InventoryFilterList",
```

```
                "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n            <p>\n                <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
            </li>\n        <li>\n            <p>\n                <code>application_name</code> - The name of the
application. \n            Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
    </li>\n        <li>\n            <p>\n                <code>license_included</code> - The type of license included. \n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n            Possible values
are <code>sql-server-enterprise</code> | \n            <code>sql-server-standard</code> | \n            <code>sql-
server-web</code> | \n            <code>windows-server-datacenter</code>.</p>\n        </li>\n        <li>\n
    <p>\n                <code>platform</code> - The platform of the resource.\n            Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n        </li>\n        <li>\n            <p>\n
        <code>resource_id</code> - The ID of the resource. \n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n            <p>\n
    <code>tag:<key></code> - The key/value combination of a tag assigned \n            to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n            <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n        </li>\n        </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
        "type": "structure",
        "members": {
            "ResourceInventoryList": {
                "target": "com.amazonaws.licensemanager#ResourceInventoryList",
                "traits": {
                    "smithy.api#documentation": "<p>Information
about the resources.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListTagsForResource": {
        "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

    "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
```

```
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the tags.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#ListTokens": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListTokensRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListTokensResponse"
  },
  "errors":
[

    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists your tokens.</p>"
  }
},
"com.amazonaws.licensemanager#ListTokensRequest": {
  "type": "structure",
  "members": {
    "TokenIds": {
      "target": "com.amazonaws.licensemanager#StringList",
      "traits": {
```

```
              "smithy.api#documentation": "<p>Token IDs.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n        <li>\n            <p>\n                <code>LicenseArns</code>\n            </p>\n         </li>\n
</ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
          "Tokens": {
            "target": "com.amazonaws.licensemanager#TokenList",
            "traits": {
              "smithy.api#documentation": "<p>Received token details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
```

```
        },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
  {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying license\n      consumption details by resource at a selected point in time. Use this action to audit the\n      current license consumption for any license inventory and configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
```

```
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#Filters",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n      are supported:</p>\n      <ul>\n         <li>\n            <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
      Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n         </li>\n
 <li>\n            <p>\n               <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n         </li>\n         <li>\n            <p>\n
   <code>resourceAccount</code> - The ID of the account that owns the resource. \n         Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n         </li>\n      </ul>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
          "type": "structure",
          "members": {

    "LicenseConfigurationUsageList": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
              "traits": {
                "smithy.api#documentation": "<p>Information about the license configurations.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
```

```
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Location": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Long": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#ManagedResourceSummary": {
      "type": "structure",
      "members": {
        "ResourceType":
{
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
          }
        },
        "AssociationCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
      }
    },
    "com.amazonaws.licensemanager#MaxSize100": {

      "type": "integer",
      "traits": {
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
```

```
"com.amazonaws.licensemanager#MaxSize3StringList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 3
        }
    }
},
"com.amazonaws.licensemanager#Message": {
    "type": "string"
},
"com.amazonaws.licensemanager#Metadata": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The key name.</p>"
            }
        },

        "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
    }
},
"com.amazonaws.licensemanager#MetadataList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#Metadata"
    }
},
"com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#Options": {
        "type": "structure",
        "members": {
          "ActivationOverrideBehavior": {
            "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
            "traits": {
              "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n        Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n        <ul>\n        <li>\n            <p>As a license administrator, you can
optionally specify an\n            <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
</li>\n        <li>\n            <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n            account in your
organization.</p>\n        </li>\n        <li>\n            <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n            <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n        are activating the grant.</p>\n        </li>\n        </ul>\n        <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n        <dd>\n            <p>Use this value to activate a grant
without replacing any member accounts\n            active grants for the same product.</p>\n        </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n        <dd>\n            <p>Use this value to activate a
grant and disable other active grants in any\n            member accounts
 for the same product. This action will also replace their\n            previously activated grants with this activated
grant.</p>\n        </dd>\n        </dl>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n        override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
      },
      "com.amazonaws.licensemanager#OrganizationConfiguration": {
        "type": "structure",
        "members": {
          "EnableIntegration": {
            "target": "com.amazonaws.licensemanager#Boolean",
            "traits": {
              "smithy.api#default": false,
              "smithy.api#documentation":
```

```
    "<p>Enables Organizations integration.</p>",
              "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
      }
    },
    "com.amazonaws.licensemanager#PrincipalArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 1
        }
      }
    },
    "com.amazonaws.licensemanager#ProductInformation": {
      "type": "structure",
      "members": {
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Resource type. The
possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductInformationFilterList": {
          "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
          "traits": {
            "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n        <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n        <p>The following filters and are supported when
the resource type \n        is <code>SSM_MANAGED</code>:</p>\n
      <ul>\n        <li>\n          <p>\n            <code>Application Name</code> - The name of the
application.\n          Logical operator is <code>EQUALS</code>.</p>\n        </li>\n        <li>\n
<p>\n            <code>Application Publisher</code> - The publisher of the application.\n          Logical
operator is <code>EQUALS</code>.</p>\n        </li>\n        <li>\n          <p>\n
<code>Application Version</code> - The version of the application.\n          Logical operator is
<code>EQUALS</code>.</p>\n        </li>\n        <li>\n          <p>\n            <code>Platform
```

Name</code> - The name of the platform.\n   Logical operator is <code>EQUALS</code>.</p>\n
\n   \n   <p>\n   <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
        \n   \n   <p>\n   <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>. The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n   </p>\n   \n   \n   <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n   Logical operator is <code>NOT_EQUALS</code>.</p>\n   \n
\n   <p>\n   <code>License Included</code> - The type of license included.\n   Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n   Possible values are:
 <code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-server-
web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n
<p>The following filters and logical operators are supported when the resource type\n   is
<code>RDS</code>:</p>\n   \n   \n   <p>\n   <code>Engine Edition</code> -
The edition of the database engine.\n   Logical operator is <code>EQUALS</code>.\n   Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n   \n   \n   <p>\n   <code>License Pack</code> - The license
pack.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>data
guard</code> | \n   <code>diagnostic pack sqlt</code> | \n
          <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n   <code>olap</code>.</p>\n
\n   ",
                "smithy.api#required": {}
              }
            }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#ProductInformationFilter": {
        "type": "structure",
        "members": {
          "ProductInformationFilterName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Filter name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductInformationFilterValue": {
            "target": "com.amazonaws.licensemanager#StringList",

            "traits": {
              "smithy.api#documentation": "<p>Filter value.</p>"
            }
          },
```

```
        "ProductInformationFilterComparator": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Logical operator.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes product information filters.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformationFilter"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformation"
      }
    },
    "com.amazonaws.licensemanager#ProvisionalConfiguration": {
      "type": "structure",
      "members": {
        "MaxTimeToLiveInMinutes": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#RateLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
```

```
            "aws.protocols#awsQueryError": {
              "code": "RateLimitExceeded",
              "httpResponseCode": 429
            },
            "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 429
          }
        },
        "com.amazonaws.licensemanager#ReceivedMetadata": {
          "type": "structure",
          "members": {
            "ReceivedStatus": {
              "target": "com.amazonaws.licensemanager#ReceivedStatus",
              "traits": {
                "smithy.api#documentation": "<p>Received status.</p>"
              }
            },
            "ReceivedStatusReason": {
              "target": "com.amazonaws.licensemanager#StatusReasonMessage",
              "traits": {
                "smithy.api#documentation": "<p>Received
status reason.</p>"
              }
            },
            "AllowedOperations": {
              "target": "com.amazonaws.licensemanager#AllowedOperationList",
              "traits": {
                "smithy.api#documentation": "<p>Allowed operations.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
          }
        },
        "com.amazonaws.licensemanager#ReceivedStatus": {
          "type": "enum",
          "members": {
            "PENDING_WORKFLOW": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "PENDING_WORKFLOW"
              }
            },
            "PENDING_ACCEPT": {
              "target": "smithy.api#Unit",
```

```
            "traits": {
                "smithy.api#enumValue":
"PENDING_ACCEPT"
            }
        },
        "REJECTED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "REJECTED"
            }
        },
        "ACTIVE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "ACTIVE"
            }
        },
        "FAILED_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "FAILED_WORKFLOW"
            }
        },
        "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "DELETED"
            }
        },
        "DISABLED": {
            "target": "smithy.api#Unit",
             "traits": {
                "smithy.api#enumValue": "DISABLED"
            }
        },
        "WORKFLOW_COMPLETED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "WORKFLOW_COMPLETED"
            }
        }
    }
},
"com.amazonaws.licensemanager#RedirectException": {
    "type": "structure",
    "members": {
        "Location": {
            "target": "com.amazonaws.licensemanager#Location",
```

```
          "traits": {
            "smithy.api#httpHeader": "Location"
          }
        },
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 308
      }
    },
    "com.amazonaws.licensemanager#RejectGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#RejectGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#RejectGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
```

```
            }
        },
        "com.amazonaws.licensemanager#RejectGrantRequest": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#RejectGrantResponse": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant ARN.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant version.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#RenewType":
{
            "type": "enum",
            "members": {
```

```
            "NONE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "None"
                }
            },
            "WEEKLY": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Weekly"
                }
            },
            "MONTHLY": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Monthly"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#ReportContext": {
        "type": "structure",
        "members": {
            "licenseConfigurationArns": {
                "target": "com.amazonaws.licensemanager#ArnList",
                "traits": {

        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportFrequency": {
        "type": "structure",
        "members": {
            "value": {
                "target": "com.amazonaws.licensemanager#Integer",
                "traits": {
                    "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n       The only supported value is <code>1</code>.</p>"
                }
            },
            "period": {
```

          "target": "com.amazonaws.licensemanager#ReportFrequencyType",

          "traits": {
            "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReportFrequencyType": {
      "type": "enum",
      "members": {
        "DAY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DAY"
          }
        },
        "WEEK": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WEEK"
          }
        },
        "MONTH": {
          "target": "smithy.api#Unit",
          "traits":
{
            "smithy.api#enumValue": "MONTH"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportGenerator": {
      "type": "structure",
      "members": {
        "ReportGeneratorName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>"
          }
        },
        "ReportType": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {

```
                    "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
                }
            },
            "ReportContext": {
                "target": "com.amazonaws.licensemanager#ReportContext",
                "traits": {
                    "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
                }
            },
            "ReportFrequency": {
                "target": "com.amazonaws.licensemanager#ReportFrequency",
                "traits": {
                    "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
                }
            },
            "LicenseManagerReportGeneratorArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
                }
            },
            "LastRunStatus": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
                }
            },
            "LastRunFailureReason":
{
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
                }
            },
            "LastReportGenerationTime": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
                }
            },
            "ReportCreatorAccount": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
                }
            },
```

```
            "Description": {
               "target": "com.amazonaws.licensemanager#String",

  "traits": {
                  "smithy.api#documentation": "<p>Description of the report generator.</p>"
               }
            },
            "S3Location": {
               "target": "com.amazonaws.licensemanager#S3Location",
               "traits": {
                  "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
               }
            },
            "CreateTime": {
               "target": "com.amazonaws.licensemanager#String",
               "traits": {
                  "smithy.api#documentation": "<p>Time the report was created.</p>"
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanager#TagList",
               "traits": {
                  "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
               }

    }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
         }
      },
      "com.amazonaws.licensemanager#ReportGeneratorList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanager#ReportGenerator"
         }
      },
      "com.amazonaws.licensemanager#ReportGeneratorName": {
         "type": "string",
         "traits": {
            "smithy.api#length": {
               "min": 1,
               "max": 100
            }
         }
      },
      "com.amazonaws.licensemanager#ReportType": {
```

```
      "type": "enum",
      "members": {
        "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
          }

    },
        "LICENSE_CONFIGURATION_USAGE_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationUsageReport"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportTypeList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportType"
      }
    },
    "com.amazonaws.licensemanager#ResourceInventory": {
      "type": "structure",
      "members": {
        "ResourceId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the resource.</p>"
          }
        },
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits":

{
            "smithy.api#documentation": "<p>Type of resource.</p>"
          }
        },
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },
        "Platform": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
          "smithy.api#documentation": "<p>Platform of the resource.</p>"
        }
      },
      "PlatformVersion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
        }
      },
      "ResourceOwningAccountId":
{
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about a resource.</p>"
    }
  },
  "com.amazonaws.licensemanager#ResourceInventoryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ResourceInventory"
    }
  },
  "com.amazonaws.licensemanager#ResourceLimitExceededException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "ResourceLimitExceeded",

        "httpResponseCode": 400
      },
      "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#ResourceNotFoundException": {
    "type": "structure",
    "members": {
```

```
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InvalidResource.NotFound",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>The resource cannot be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#ResourceType": {
      "type": "enum",
      "members": {
        "EC2_INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_INSTANCE"
          }
        },
        "EC2_HOST": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_HOST"
          }
        },
        "EC2_AMI": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_AMI"
          }
        },
        "RDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "RDS"
          }
        },
        "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
          }
        }
      }
```

```
    },
    "com.amazonaws.licensemanager#S3Location": {
      "type": "structure",
      "members": {
        "bucket": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
          }
        },
        "keyPrefix": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
      }
    },
    "com.amazonaws.licensemanager#ServerInternalException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InternalError",
          "httpResponseCode": 500
        },
        "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licensemanager#SignedToken": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 4096
        }
      }
    },
    "com.amazonaws.licensemanager#StatusReasonMessage": {
```

```
            "type": "string",
            "traits":
{
                "smithy.api#length": {
                    "min": 0,
                    "max": 400
                },
                "smithy.api#pattern": "^[\\s\\S]+$"
            }
        },
        "com.amazonaws.licensemanager#String": {
            "type": "string"
        },
        "com.amazonaws.licensemanager#StringList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#String"
            }
        },
        "com.amazonaws.licensemanager#Tag": {
            "type": "structure",
            "members": {
                "Key": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Tag key.</p>"
                    }
                },
                "Value": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation":
"<p>Tag value.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#TagKeyList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#String"
            }
        },
        "com.amazonaws.licensemanager#TagList": {
            "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanager#Tag"
        }
      },
      "com.amazonaws.licensemanager#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#TagResourceResponse"
        },
        "errors": [
          {
            "target":
"com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits":
{
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>One or more tags.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#TokenData": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>"
            }
          },
          "TokenType": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "ExpirationTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
format.</p>"
            }
          },
          "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
```

```
        "traits": {
          "smithy.api#documentation": "<p>Data specified by the caller.</p>"
        }
      },
      "RoleArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
        }
      }
    },
    "traits": {

      "smithy.api#documentation": "<p>Describes a token.</p>"
    }
  },
  "com.amazonaws.licensemanager#TokenList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#TokenData"
    }
  },
  "com.amazonaws.licensemanager#TokenString": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 4096
      },
      "smithy.api#pattern": "^\\S+$"
    }
  },
  "com.amazonaws.licensemanager#TokenType": {
    "type": "enum",
    "members": {
      "REFRESH_TOKEN": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "REFRESH_TOKEN"
        }
```

```
          }
        }
      },
      "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
  {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
        }
      },
```

```
"com.amazonaws.licensemanager#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licensemanager#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
    },
    "output": {

        "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseConfigurationStatus": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
          "traits": {
            "smithy.api#documentation": "<p>New status of the license configuration.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {

    "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
  configuration is licenseAffinityToHost.</p>"
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
```

```
              "target": "com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
              }
            },
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>New description of the license configuration.</p>"
              }
            },
            "ProductInformationList": {
              "target": "com.amazonaws.licensemanager#ProductInformationList",
              "traits": {
                "smithy.api#documentation": "<p>New product information.</p>"
              }
            },
            "DisassociateWhenNotFound": {
              "target": "com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
              }
            }
          },
          "traits":
{
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
```

```
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
 {
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
            "smithy.api#required": {}
          }
        },
        "ReportGeneratorName": {
```

```
              "target": "com.amazonaws.licensemanager#ReportGeneratorName",
              "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>",
                "smithy.api#required": {}
              }
            },
            "Type": {
              "target": "com.amazonaws.licensemanager#ReportTypeList",
              "traits": {
                "smithy.api#documentation":
```

"<p>Type of reports to generate. The following report types are supported:</p>\n   \n   \n   <p>License configuration report - Reports the number and details of consumed licenses for a license configuration.</p>\n   \n   \n   <p>Resource report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n   \n   ",

```
                "smithy.api#required": {}
              }
            },
            "ReportContext": {
              "target": "com.amazonaws.licensemanager#ReportContext",
              "traits": {
                "smithy.api#documentation": "<p>The report context.</p>",
                "smithy.api#required": {}
              }
            },
            "ReportFrequency": {
              "target": "com.amazonaws.licensemanager#ReportFrequency",
              "traits":
              {
                "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                "smithy.api#required": {}
              }
            },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientRequestToken",
              "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
              }
            }
          },
          "traits": {
```

```
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
  {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },

          {
            "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
          },
          {
            "target": "com.amazonaws.licensemanager#LicenseUsageException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
```

```
        }
      },

    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
              "smithy.api#required": {}
            }
          },
          "AddLicenseSpecifications": {
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
              "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
            }
          },
          "RemoveLicenseSpecifications": {
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
              "smithy.api#documentation":
  "<p>ARNs of the license configurations to remove.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
    "com.amazonaws.licensemanager#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
        },
        "errors": [
```

```
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
     },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
    }
  },
  "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "S3BucketArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
        }
      },
      "SnsTopicArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
        }
      },
      "OrganizationConfiguration": {
        "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
        }
      },
      "EnableCrossAccountsDiscovery": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
```

```
                    "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

              }
                }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
      "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
      "com.amazonaws.licensemanager#UsageOperation": {
          "type": "string",
          "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            }
          }
        },
      "com.amazonaws.licensemanager#ValidationException": {
          "type": "structure",
          "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

        "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        }
      }
  }
}
```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"

```go
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
	if params == nil {
		params = &DeleteLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*DeleteLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type DeleteLicenseInput struct {

	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// Current version
	of the license.
	//
	// This member is required.
	SourceVersion *string

	noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

	// Date when the license is deleted.
	DeletionDate *string

	// License status.
	Status types.LicenseDeletionStatus
```

```go
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack,
    options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
        return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
        return err
    }
```

```
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if
 err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "DeleteLicense",
 }
}
```

1.30 aws-aws-sdk-go 1.23.4

1.30.1 Available under license :

```
{
 "version": "1.0",
 "examples": {
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
{
 "version":"2.0",
 "metadata":{
  "apiVersion":"2018-08-01",
  "endpointPrefix":"license-manager",
  "jsonVersion":"1.1",
  "protocol":"json",
  "serviceFullName":"AWS License Manager",
  "serviceId":"License Manager",
  "signatureVersion":"v4",
  "targetPrefix":"AWSLicenseManager",
  "uid":"license-manager-2018-08-01"
 },
 "operations":{
  "CreateLicenseConfiguration":{
   "name":"CreateLicenseConfiguration",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"CreateLicenseConfigurationRequest"},
   "output":{"shape":"CreateLicenseConfigurationResponse"},
   "errors":[
    {"shape":"InvalidParameterValueException"},
    {"shape":"ServerInternalException"},
    {"shape":"ResourceLimitExceededException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"}
   ]
  },
  "DeleteLicenseConfiguration":{
   "name":"DeleteLicenseConfiguration",
   "http":{
    "method":"POST",
```

```
   "requestUri":"/"
  },
  "input":{"shape":"DeleteLicenseConfigurationRequest"},
  "output":{"shape":"DeleteLicenseConfigurationResponse"},
  "errors":[
   {"shape":"InvalidParameterValueException"},
   {"shape":"ServerInternalException"},
   {"shape":"AuthorizationException"},
   {"shape":"AccessDeniedException"},
   {"shape":"RateLimitExceededException"}
  ]
 },
 "GetLicenseConfiguration":{
  "name":"GetLicenseConfiguration",
  "http":{
   "method":"POST",
   "requestUri":"/"
  },
  "input":{"shape":"GetLicenseConfigurationRequest"},
  "output":{"shape":"GetLicenseConfigurationResponse"},
  "errors":[
   {"shape":"InvalidParameterValueException"},
   {"shape":"ServerInternalException"},
   {"shape":"AuthorizationException"},
   {"shape":"AccessDeniedException"},
   {"shape":"RateLimitExceededException"}
  ]
 },
 "GetServiceSettings":{
  "name":"GetServiceSettings",
  "http":{
   "method":"POST",
   "requestUri":"/"
  },
  "input":{"shape":"GetServiceSettingsRequest"},
  "output":{"shape":"GetServiceSettingsResponse"},
  "errors":[
   {"shape":"ServerInternalException"},
   {"shape":"AuthorizationException"},
   {"shape":"AccessDeniedException"},
   {"shape":"RateLimitExceededException"}
  ]
 },
 "ListAssociationsForLicenseConfiguration":{
  "name":"ListAssociationsForLicenseConfiguration",
  "http":{
   "method":"POST",
   "requestUri":"/"
```

```
      },
      "input":{"shape":"ListAssociationsForLicenseConfigurationRequest"},
      "output":{"shape":"ListAssociationsForLicenseConfigurationResponse"},
      "errors":[
        {"shape":"InvalidParameterValueException"},
        {"shape":"FilterLimitExceededException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    },
    "ListLicenseConfigurations":{
      "name":"ListLicenseConfigurations",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"ListLicenseConfigurationsRequest"},
      "output":{"shape":"ListLicenseConfigurationsResponse"},
      "errors":[
        {"shape":"InvalidParameterValueException"},
        {"shape":"ServerInternalException"},
        {"shape":"FilterLimitExceededException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    },
    "ListLicenseSpecificationsForResource":{
      "name":"ListLicenseSpecificationsForResource",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"ListLicenseSpecificationsForResourceRequest"},
      "output":{"shape":"ListLicenseSpecificationsForResourceResponse"},
      "errors":[
        {"shape":"InvalidParameterValueException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    },
    "ListResourceInventory":{
      "name":"ListResourceInventory",
      "http":{
```

```
   "method":"POST",
    "requestUri":"/"
  },
  "input":{"shape":"ListResourceInventoryRequest"},
  "output":{"shape":"ListResourceInventoryResponse"},
  "errors":[
   {"shape":"InvalidParameterValueException"},
   {"shape":"ServerInternalException"},
   {"shape":"FilterLimitExceededException"},
   {"shape":"FailedDependencyException"},
   {"shape":"AuthorizationException"},
   {"shape":"AccessDeniedException"},
   {"shape":"RateLimitExceededException"}
  ]
 },
 "ListTagsForResource":{
  "name":"ListTagsForResource",
  "http":{
   "method":"POST",
    "requestUri":"/"
  },
  "input":{"shape":"ListTagsForResourceRequest"},
  "output":{"shape":"ListTagsForResourceResponse"},
  "errors":[
   {"shape":"InvalidParameterValueException"},
   {"shape":"ServerInternalException"},
   {"shape":"AuthorizationException"},
   {"shape":"AccessDeniedException"},
   {"shape":"RateLimitExceededException"}
  ]
 },
 "ListUsageForLicenseConfiguration":{
  "name":"ListUsageForLicenseConfiguration",
  "http":{
   "method":"POST",
   "requestUri":"/"
  },
  "input":{"shape":"ListUsageForLicenseConfigurationRequest"},
  "output":{"shape":"ListUsageForLicenseConfigurationResponse"},
  "errors":[
   {"shape":"InvalidParameterValueException"},
   {"shape":"FilterLimitExceededException"},
   {"shape":"ServerInternalException"},
   {"shape":"AuthorizationException"},
   {"shape":"AccessDeniedException"},
   {"shape":"RateLimitExceededException"}
  ]
 },
```

```
"TagResource":{
 "name":"TagResource",
 "http":{
   "method":"POST",
   "requestUri":"/"
 },
 "input":{"shape":"TagResourceRequest"},
 "output":{"shape":"TagResourceResponse"},
 "errors":[
   {"shape":"InvalidParameterValueException"},
   {"shape":"ServerInternalException"},
   {"shape":"AuthorizationException"},
   {"shape":"AccessDeniedException"},
   {"shape":"RateLimitExceededException"}
 ]
},
"UntagResource":{
 "name":"UntagResource",
 "http":{
   "method":"POST",
   "requestUri":"/"
 },
 "input":{"shape":"UntagResourceRequest"},
 "output":{"shape":"UntagResourceResponse"},
 "errors":[
   {"shape":"InvalidParameterValueException"},
   {"shape":"ServerInternalException"},
   {"shape":"AuthorizationException"},
   {"shape":"AccessDeniedException"},
   {"shape":"RateLimitExceededException"}
 ]
},
"UpdateLicenseConfiguration":{
  "name":"UpdateLicenseConfiguration",
 "http":{
   "method":"POST",
   "requestUri":"/"
 },
 "input":{"shape":"UpdateLicenseConfigurationRequest"},
 "output":{"shape":"UpdateLicenseConfigurationResponse"},
 "errors":[
   {"shape":"InvalidParameterValueException"},
   {"shape":"ServerInternalException"},
   {"shape":"AuthorizationException"},
   {"shape":"AccessDeniedException"},
   {"shape":"RateLimitExceededException"}
 ]
},
```

```
"UpdateLicenseSpecificationsForResource":{
  "name":"UpdateLicenseSpecificationsForResource",
  "http":{
    "method":"POST",
    "requestUri":"/"
  },
  "input":{"shape":"UpdateLicenseSpecificationsForResourceRequest"},
  "output":{"shape":"UpdateLicenseSpecificationsForResourceResponse"},
  "errors":[
    {"shape":"InvalidParameterValueException"},
    {"shape":"InvalidResourceStateException"},
    {"shape":"LicenseUsageException"},
    {"shape":"ServerInternalException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"}
  ]
},
"UpdateServiceSettings":{
  "name":"UpdateServiceSettings",
  "http":{
    "method":"POST",
    "requestUri":"/"
  },
  "input":{"shape":"UpdateServiceSettingsRequest"},
  "output":{"shape":"UpdateServiceSettingsResponse"},
  "errors":[
    {"shape":"InvalidParameterValueException"},
    {"shape":"ServerInternalException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"}
  ]
}
},
"shapes":{
  "AccessDeniedException":{
    "type":"structure",
    "members":{
      "Message":{"shape":"Message"}
    },
    "exception":true
  },
  "AuthorizationException":{
    "type":"structure",
    "members":{
      "Message":{"shape":"Message"}
```

```
    },
    "exception":true
  },
  "Boolean":{"type":"boolean"},
  "BoxBoolean":{"type":"boolean"},
  "BoxInteger":{"type":"integer"},
  "BoxLong":{"type":"long"},
  "ConsumedLicenseSummary":{
   "type":"structure",
   "members":{
    "ResourceType":{"shape":"ResourceType"},
    "ConsumedLicenses":{"shape":"BoxLong"}
   }
  },
  "ConsumedLicenseSummaryList":{
   "type":"list",
   "member":{"shape":"ConsumedLicenseSummary"}
  },
  "CreateLicenseConfigurationRequest":{
   "type":"structure",
   "required":[
    "Name",
    "LicenseCountingType"
   ],
   "members":{
    "Name":{"shape":"String"},
    "Description":{"shape":"String"},
    "LicenseCountingType":{"shape":"LicenseCountingType"},
    "LicenseCount":{"shape":"BoxLong"},
    "LicenseCountHardLimit":{"shape":"BoxBoolean"},
    "LicenseRules":{"shape":"StringList"},
    "Tags":{"shape":"TagList"}
   }
  },
  "CreateLicenseConfigurationResponse":{
   "type":"structure",
   "members":{
    "LicenseConfigurationArn":{"shape":"String"}
   }
  },
  "DateTime":{"type":"timestamp"},
  "DeleteLicenseConfigurationRequest":{
   "type":"structure",
   "required":["LicenseConfigurationArn"],
   "members":{
    "LicenseConfigurationArn":{"shape":"String"}
   }
  },
```

```
"DeleteLicenseConfigurationResponse":{
 "type":"structure",
 "members":{
 }
},
"FailedDependencyException":{
 "type":"structure",
 "members":{
  "Message":{"shape":"Message"}
 },
 "exception":true,
 "fault":true
},
"Filter":{
 "type":"structure",
 "members":{
  "Name":{"shape":"FilterName"},
  "Values":{"shape":"FilterValues"}
 }
},
"FilterLimitExceededException":{
 "type":"structure",
 "members":{
  "Message":{"shape":"Message"}
 },
 "exception":true
},
"FilterName":{"type":"string"},
 "FilterValue":{"type":"string"},
"FilterValues":{
 "type":"list",
 "member":{"shape":"FilterValue"}
},
"Filters":{
 "type":"list",
 "member":{"shape":"Filter"}
},
"GetLicenseConfigurationRequest":{
 "type":"structure",
 "required":["LicenseConfigurationArn"],
 "members":{
  "LicenseConfigurationArn":{"shape":"String"}
 }
},
"GetLicenseConfigurationResponse":{
 "type":"structure",
 "members":{
  "LicenseConfigurationId":{"shape":"String"},
```

      "LicenseConfigurationArn":{"shape":"String"},
      "Name":{"shape":"String"},
      "Description":{"shape":"String"},
      "LicenseCountingType":{"shape":"LicenseCountingType"},
      "LicenseRules":{"shape":"StringList"},
      "LicenseCount":{"shape":"BoxLong"},
      "LicenseCountHardLimit":{"shape":"BoxBoolean"},
      "ConsumedLicenses":{"shape":"BoxLong"},
      "Status":{"shape":"String"},
      "OwnerAccountId":{"shape":"String"},
      "ConsumedLicenseSummaryList":{"shape":"ConsumedLicenseSummaryList"},
      "ManagedResourceSummaryList":{"shape":"ManagedResourceSummaryList"},
      "Tags":{"shape":"TagList"}
     }
    },
    "GetServiceSettingsRequest":{
     "type":"structure",
     "members":{
     }
    },
    "GetServiceSettingsResponse":{
     "type":"structure",
     "members":{
      "S3BucketArn":{"shape":"String"},
      "SnsTopicArn":{"shape":"String"},
      "OrganizationConfiguration":{"shape":"OrganizationConfiguration"},
      "EnableCrossAccountsDiscovery":{"shape":"BoxBoolean"}
     }
    },
    "InvalidParameterValueException":{
     "type":"structure",
     "members":{
      "Message":{"shape":"Message"}
     },
     "exception":true,
     "synthetic":true
    },
    "InvalidResourceStateException":{
     "type":"structure",
     "members":{
      "Message":{"shape":"Message"}
     },
     "exception":true
    },
    "InventoryFilter":{
     "type":"structure",
     "required":[
      "Name",

```
    "Condition"
   ],
   "members":{
    "Name":{"shape":"String"},
    "Condition":{"shape":"InventoryFilterCondition"},
    "Value":{"shape":"String"}
   }
  },
  "InventoryFilterCondition":{
   "type":"string",
   "enum":[
    "EQUALS",
    "NOT_EQUALS",
    "BEGINS_WITH",
    "CONTAINS"
   ]
  },
  "InventoryFilterList":{
   "type":"list",
   "member":{"shape":"InventoryFilter"}
  },
  "LicenseConfiguration":{
   "type":"structure",
   "members":{
    "LicenseConfigurationId":{"shape":"String"},
    "LicenseConfigurationArn":{"shape":"String"},
    "Name":{"shape":"String"},
    "Description":{"shape":"String"},
    "LicenseCountingType":{"shape":"LicenseCountingType"},
    "LicenseRules":{"shape":"StringList"},
    "LicenseCount":{"shape":"BoxLong"},
    "LicenseCountHardLimit":{"shape":"BoxBoolean"},
    "ConsumedLicenses":{"shape":"BoxLong"},
    "Status":{"shape":"String"},
    "OwnerAccountId":{"shape":"String"},
    "ConsumedLicenseSummaryList":{"shape":"ConsumedLicenseSummaryList"},
    "ManagedResourceSummaryList":{"shape":"ManagedResourceSummaryList"}
   }
  },
  "LicenseConfigurationAssociation":{
   "type":"structure",
   "members":{
    "ResourceArn":{"shape":"String"},
    "ResourceType":{"shape":"ResourceType"},
    "ResourceOwnerId":{"shape":"String"},
    "AssociationTime":{"shape":"DateTime"}
   }
  },
```

```
"LicenseConfigurationAssociations":{
 "type":"list",
 "member":{"shape":"LicenseConfigurationAssociation"}
},
"LicenseConfigurationStatus":{
 "type":"string",
 "enum":[
  "AVAILABLE",
  "DISABLED"
 ]
},
"LicenseConfigurationUsage":{
 "type":"structure",
 "members":{
  "ResourceArn":{"shape":"String"},
  "ResourceType":{"shape":"ResourceType"},
  "ResourceStatus":{"shape":"String"},
  "ResourceOwnerId":{"shape":"String"},
  "AssociationTime":{"shape":"DateTime"},
  "ConsumedLicenses":{"shape":"BoxLong"}
 }
},
"LicenseConfigurationUsageList":{
 "type":"list",
 "member":{"shape":"LicenseConfigurationUsage"}
},
"LicenseConfigurations":{
 "type":"list",
 "member":{"shape":"LicenseConfiguration"}
},
"LicenseCountingType":{
 "type":"string",
 "enum":[
  "vCPU",
  "Instance",
  "Core",
  "Socket"
 ]
},
"LicenseSpecification":{
 "type":"structure",
 "required":["LicenseConfigurationArn"],
 "members":{
  "LicenseConfigurationArn":{"shape":"String"}
 }
},
"LicenseSpecifications":{
 "type":"list",
```

```
      "member":{"shape":"LicenseSpecification"}
     },
     "LicenseUsageException":{
      "type":"structure",
      "members":{
       "Message":{"shape":"Message"}
      },
      "exception":true
     },


     "ListAssociationsForLicenseConfigurationRequest":{
      "type":"structure",
      "required":["LicenseConfigurationArn"],
      "members":{
       "LicenseConfigurationArn":{"shape":"String"},
       "MaxResults":{"shape":"BoxInteger"},
       "NextToken":{"shape":"String"}
      }
     },
     "ListAssociationsForLicenseConfigurationResponse":{
      "type":"structure",
      "members":{
       "LicenseConfigurationAssociations":{"shape":"LicenseConfigurationAssociations"},
       "NextToken":{"shape":"String"}
      }
     },
     "ListLicenseConfigurationsRequest":{
      "type":"structure",
      "members":{
       "LicenseConfigurationArns":{"shape":"StringList"},
       "MaxResults":{"shape":"BoxInteger"},
       "NextToken":{"shape":"String"},
       "Filters":{"shape":"Filters"}
      }
     },
     "ListLicenseConfigurationsResponse":{
      "type":"structure",
      "members":{
       "LicenseConfigurations":{"shape":"LicenseConfigurations"},
       "NextToken":{"shape":"String"}
      }
     },
     "ListLicenseSpecificationsForResourceRequest":{
      "type":"structure",
      "required":["ResourceArn"],
      "members":{
       "ResourceArn":{"shape":"String"},
       "MaxResults":{"shape":"BoxInteger"},
```

```
    "NextToken":{"shape":"String"}
   }
  },
  "ListLicenseSpecificationsForResourceResponse":{
   "type":"structure",
   "members":{
    "LicenseSpecifications":{"shape":"LicenseSpecifications"},
    "NextToken":{"shape":"String"}
   }
  },
  "ListResourceInventoryRequest":{
   "type":"structure",
   "members":{
    "MaxResults":{"shape":"BoxInteger"},
    "NextToken":{"shape":"String"},
    "Filters":{"shape":"InventoryFilterList"}
   }
  },
  "ListResourceInventoryResponse":{
   "type":"structure",
   "members":{
    "ResourceInventoryList":{"shape":"ResourceInventoryList"},
    "NextToken":{"shape":"String"}
   }
  },
  "ListTagsForResourceRequest":{
   "type":"structure",
   "required":["ResourceArn"],
   "members":{
    "ResourceArn":{"shape":"String"}
   }
  },
  "ListTagsForResourceResponse":{
   "type":"structure",
   "members":{
    "Tags":{"shape":"TagList"}
   }
  },
  "ListUsageForLicenseConfigurationRequest":{
   "type":"structure",
   "required":["LicenseConfigurationArn"],
   "members":{
    "LicenseConfigurationArn":{"shape":"String"},
    "MaxResults":{"shape":"BoxInteger"},
    "NextToken":{"shape":"String"},
    "Filters":{"shape":"Filters"}
   }
  },
```

```
"ListUsageForLicenseConfigurationResponse":{
 "type":"structure",
 "members":{
   "LicenseConfigurationUsageList":{"shape":"LicenseConfigurationUsageList"},
   "NextToken":{"shape":"String"}
 }
},
"ManagedResourceSummary":{
 "type":"structure",
 "members":{
   "ResourceType":{"shape":"ResourceType"},
   "AssociationCount":{"shape":"BoxLong"}
 }
},
"ManagedResourceSummaryList":{
  "type":"list",
 "member":{"shape":"ManagedResourceSummary"}
},
"Message":{"type":"string"},
"OrganizationConfiguration":{
 "type":"structure",
 "required":["EnableIntegration"],
 "members":{
   "EnableIntegration":{"shape":"Boolean"}
 }
},
"RateLimitExceededException":{
 "type":"structure",
 "members":{
   "Message":{"shape":"Message"}
 },
 "exception":true
},
"ResourceInventory":{
 "type":"structure",
 "members":{
   "ResourceId":{"shape":"String"},
   "ResourceType":{"shape":"ResourceType"},
   "ResourceArn":{"shape":"String"},
   "Platform":{"shape":"String"},
   "PlatformVersion":{"shape":"String"},
   "ResourceOwningAccountId":{"shape":"String"}
 }
},
"ResourceInventoryList":{
 "type":"list",
 "member":{"shape":"ResourceInventory"}
},
```

```
"ResourceLimitExceededException":{
 "type":"structure",
  "members":{
   "Message":{"shape":"Message"}
  },
  "exception":true
},
"ResourceType":{
 "type":"string",
 "enum":[
   "EC2_INSTANCE",
   "EC2_HOST",
   "EC2_AMI"
 ]
},
"ServerInternalException":{
 "type":"structure",
 "members":{
   "Message":{"shape":"Message"}
 },
 "exception":true,
 "fault":true
},
"String":{"type":"string"},
"StringList":{
 "type":"list",
 "member":{"shape":"String"}
},
"Tag":{
 "type":"structure",
 "members":{
   "Key":{"shape":"String"},
   "Value":{"shape":"String"}
 }
},
"TagKeyList":{
 "type":"list",
 "member":{"shape":"String"}
},
"TagList":{
 "type":"list",
 "member":{"shape":"Tag"}
},
"TagResourceRequest":{
 "type":"structure",
 "required":[
   "ResourceArn",
   "Tags"
```

```
    ],
    "members":{
      "ResourceArn":{"shape":"String"},

"Tags":{"shape":"TagList"}
    }
  },
  "TagResourceResponse":{
   "type":"structure",
   "members":{
   }
  },
  "UntagResourceRequest":{
   "type":"structure",
   "required":[
     "ResourceArn",
     "TagKeys"
   ],
   "members":{
     "ResourceArn":{"shape":"String"},
     "TagKeys":{"shape":"TagKeyList"}
   }
  },
  "UntagResourceResponse":{
   "type":"structure",
   "members":{
   }
  },
  "UpdateLicenseConfigurationRequest":{
   "type":"structure",
   "required":["LicenseConfigurationArn"],
   "members":{
     "LicenseConfigurationArn":{"shape":"String"},
     "LicenseConfigurationStatus":{"shape":"LicenseConfigurationStatus"},
     "LicenseRules":{"shape":"StringList"},
     "LicenseCount":{"shape":"BoxLong"},
     "LicenseCountHardLimit":{"shape":"BoxBoolean"},
     "Name":{"shape":"String"},
     "Description":{"shape":"String"}
   }
  },
  "UpdateLicenseConfigurationResponse":{
   "type":"structure",
   "members":{
   }
  },
  "UpdateLicenseSpecificationsForResourceRequest":{
   "type":"structure",
```

```
      "required":["ResourceArn"],
      "members":{
        "ResourceArn":{"shape":"String"},
        "AddLicenseSpecifications":{"shape":"LicenseSpecifications"},
        "RemoveLicenseSpecifications":{"shape":"LicenseSpecifications"}
      }
    },
    "UpdateLicenseSpecificationsForResourceResponse":{
      "type":"structure",
      "members":{
      }
    },
    "UpdateServiceSettingsRequest":{
      "type":"structure",
      "members":{
        "S3BucketArn":{"shape":"String"},
        "SnsTopicArn":{"shape":"String"},
        "OrganizationConfiguration":{"shape":"OrganizationConfiguration"},
        "EnableCrossAccountsDiscovery":{"shape":"BoxBoolean"}
      }
    },
    "UpdateServiceSettingsResponse":{
      "type":"structure",
      "members":{
      }
    }
  }
}
{
 "version": "2.0",
 "service": "<fullname> AWS License Manager </fullname> <p> <i>This is the AWS License Manager API Reference.</i> It provides descriptions, syntax, and usage examples for each of the actions and data types for License Manager. The topic for each action shows the Query API request parameters and the XML response. You can also view the XML request elements in the WSDL. </p> <p> Alternatively, you can use one of the AWS SDKs to access an API that's tailored to the programming language or platform that you're using. For more information, see <a href=\"http://aws.amazon.com/tools/#SDKs\">AWS SDKs</a>. </p>",
 "operations": {
   "CreateLicenseConfiguration": "<p>Creates a new license configuration object. A license configuration is an abstraction of a customer license agreement that can be consumed and enforced by License Manager. Components include specifications for the license type (licensing by instance, socket, CPU, or VCPU), tenancy (shared tenancy, Amazon EC2 Dedicated
 Instance, Amazon EC2 Dedicated Host, or any of these), host affinity (how long a VM must be associated with a host), the number of licenses purchased and used.</p>",
   "DeleteLicenseConfiguration": "<p>Deletes an existing license configuration. This action fails if the configuration is in use.</p>",
   "GetLicenseConfiguration": "<p>Returns a detailed description of a license configuration.</p>",
   "GetServiceSettings": "<p>Gets License Manager settings for a region. Exposes the configured S3 bucket, SNS topic, etc., for inspection. </p>",
```

"ListAssociationsForLicenseConfiguration": "<p>Lists the resource associations for a license configuration. Resource associations need not consume licenses from a license configuration. For example, an AMI or a stopped instance may not consume a license (depending on the license rules). Use this operation to find all resources associated with a license configuration.</p>",
   "ListLicenseConfigurations": "<p>Lists license configuration
 objects for an account, each containing the name, description, license type, and other license terms modeled from a license agreement.</p>",
   "ListLicenseSpecificationsForResource": "<p>Returns the license configuration for a resource.</p>",
   "ListResourceInventory": "<p>Returns a detailed list of resources.</p>",
   "ListTagsForResource": "<p>Lists tags attached to a resource.</p>",
   "ListUsageForLicenseConfiguration": "<p>Lists all license usage records for a license configuration, displaying license consumption details by resource at a selected point in time. Use this action to audit the current license consumption for any license inventory and configuration.</p>",
   "TagResource": "<p>Attach one of more tags to any resource.</p>",
   "UntagResource": "<p>Remove tags from a resource.</p>",
   "UpdateLicenseConfiguration": "<p>Modifies the attributes of an existing license configuration object. A license configuration is an abstraction of a customer license agreement
 that can be consumed and enforced by License Manager. Components include specifications for the license type (Instances, cores, sockets, VCPUs), tenancy (shared or Dedicated Host), host affinity (how long a VM is associated with a host), the number of licenses purchased and used.</p>",
   "UpdateLicenseSpecificationsForResource": "<p>Adds or removes license configurations for a specified AWS resource. This operation currently supports updating the license specifications of AMIs, instances, and hosts. Launch templates and AWS CloudFormation templates are not managed from this operation as those resources send the license configurations directly to a resource creation operation, such as <code>RunInstances</code>.</p>",
   "UpdateServiceSettings": "<p>Updates License Manager service settings.</p>"
 },
 "shapes": {
  "AccessDeniedException": {
   "base": "<p>Access to resource denied.</p>",
   "refs": {
   }
  },
  "AuthorizationException": {
   "base": "<p>The AWS
 user account does not have permission to perform the action. Check the IAM policy associated with this account.</p>",
   "refs": {
   }
  },
  "Boolean": {
   "base": null,
   "refs": {
    "OrganizationConfiguration$EnableIntegration": "<p>Flag to activate AWS Organization integration.</p>"
   }
  },
  "BoxBoolean": {
   "base": null,
   "refs": {

"CreateLicenseConfigurationRequest$LicenseCountHardLimit": "<p>Flag indicating whether hard or soft license enforcement is used. Exceeding a hard limit results in the blocked deployment of new instances.</p>",
    "GetLicenseConfigurationResponse$LicenseCountHardLimit": "<p>Sets the number of available licenses as a hard limit.</p>",
    "GetServiceSettingsResponse$EnableCrossAccountsDiscovery": "<p>Indicates whether cross-account discovery has been enabled.</p>",
    "LicenseConfiguration$LicenseCountHardLimit": "<p>Sets the number of available licenses as a hard limit.</p>",
    "UpdateLicenseConfigurationRequest$LicenseCountHardLimit":
 "<p>Sets the number of available licenses as a hard limit.</p>",
    "UpdateServiceSettingsRequest$EnableCrossAccountsDiscovery": "<p>Activates cross-account discovery.</p>"
    }
  },
  "BoxInteger": {
   "base": null,
   "refs": {
    "ListAssociationsForLicenseConfigurationRequest$MaxResults": "<p>Maximum number of results to return in a single call. To retrieve the remaining results, make another call with the returned <code>NextToken</code> value.</p>",
    "ListLicenseConfigurationsRequest$MaxResults": "<p>Maximum number of results to return in a single call. To retrieve the remaining results, make another call with the returned <code>NextToken</code> value.</p>",
    "ListLicenseSpecificationsForResourceRequest$MaxResults": "<p>Maximum number of results to return in a single call. To retrieve the remaining results, make another call with the returned <code>NextToken</code> value.</p>",
    "ListResourceInventoryRequest$MaxResults":
 "<p>Maximum number of results to return in a single call. To retrieve the remaining results, make another call with the returned <code>NextToken</code> value.</p>",
    "ListUsageForLicenseConfigurationRequest$MaxResults": "<p>Maximum number of results to return in a single call. To retrieve the remaining results, make another call with the returned <code>NextToken</code> value.</p>"
    }
  },
  "BoxLong": {
   "base": null,
   "refs": {
    "ConsumedLicenseSummary$ConsumedLicenses": "<p>Number of licenses consumed by a resource.</p>",
    "CreateLicenseConfigurationRequest$LicenseCount": "<p>Number of licenses managed by the license configuration.</p>",
    "GetLicenseConfigurationResponse$LicenseCount": "<p>Number of available licenses.</p>",
    "GetLicenseConfigurationResponse$ConsumedLicenses": "<p>Number of licenses assigned to resources.</p>",
    "LicenseConfiguration$LicenseCount": "<p>Number of licenses managed by the license configuration.</p>",
    "LicenseConfiguration$ConsumedLicenses": "<p>Number of licenses consumed. </p>",
    "LicenseConfigurationUsage$ConsumedLicenses": "<p>Number of licenses consumed out of the total provisioned in the license configuration.</p>",
    "ManagedResourceSummary$AssociationCount": "<p>Number of resources associated with licenses.</p>",
    "UpdateLicenseConfigurationRequest$LicenseCount": "<p>New number of licenses managed by the license

configuration.</p>"
      }
    },
    "ConsumedLicenseSummary": {
      "base": "<p>Details about license consumption.</p>",
      "refs": {
        "ConsumedLicenseSummaryList$member": null
      }
    },
    "ConsumedLicenseSummaryList": {
      "base": null,
      "refs": {
        "GetLicenseConfigurationResponse$ConsumedLicenseSummaryList": "<p>List of summaries for consumed
licenses used by various resources.</p>",
        "LicenseConfiguration$ConsumedLicenseSummaryList": "<p>List of
 summaries for licenses consumed by various resources.</p>"
      }
    },
    "CreateLicenseConfigurationRequest": {
      "base": null,
      "refs": {
      }
    },
    "CreateLicenseConfigurationResponse": {
      "base": null,
      "refs": {
      }
    },
    "DateTime": {
      "base": null,
      "refs": {
        "LicenseConfigurationAssociation$AssociationTime": "<p>Time when the license configuration was associated
with the resource.</p>",
        "LicenseConfigurationUsage$AssociationTime": "<p>Time when the license configuration was initially
associated with a resource.</p>"
      }
    },
    "DeleteLicenseConfigurationRequest": {
      "base": null,
      "refs": {
      }
    },
    "DeleteLicenseConfigurationResponse": {
      "base": null,
      "refs": {
      }
    },
    "FailedDependencyException": {

"base": "<p>A dependency required to run the API is missing.</p>",
   "refs": {
   }
  },
  "Filter": {
   "base": "<p>A filter name and value pair
 that is used to return a more specific list of results from a describe operation. Filters can be used to match a set of
resources by specific criteria, such as tags, attributes, or IDs. The filters supported by a <code>Describe</code>
operation are documented with the <code>Describe</code> operation.</p>",
   "refs": {
    "Filters$member": null
   }
  },
  "FilterLimitExceededException": {
   "base": "<p>The request uses too many filters or too many filter values.</p>",
   "refs": {
   }
  },
  "FilterName": {
   "base": null,
   "refs": {
    "Filter$Name": "<p>Name of the filter. Filter names are case-sensitive.</p>"
   }
  },
  "FilterValue": {
   "base": null,
   "refs": {
    "FilterValues$member": null
   }
  },
  "FilterValues": {
   "base": null,
   "refs": {
    "Filter$Values": "<p>One or more filter values. Filter values are case-sensitive.</p>"
   }
  },
  "Filters": {
   "base": null,

   "refs": {
    "ListLicenseConfigurationsRequest$Filters": "<p>One or more filters.</p>",
    "ListUsageForLicenseConfigurationRequest$Filters": "<p>List of filters to apply.</p>"
   }
  },
  "GetLicenseConfigurationRequest": {
   "base": null,
   "refs": {
   }

```
    },
    "GetLicenseConfigurationResponse": {
     "base": null,
     "refs": {
     }
    },
    "GetServiceSettingsRequest": {
     "base": null,
     "refs": {
     }
    },
    "GetServiceSettingsResponse": {
     "base": null,
     "refs": {
     }
    },
    "InvalidParameterValueException": {
     "base": "<p>One or more parameter values are not valid.</p>",
     "refs": {
     }
    },
    "InvalidResourceStateException": {
     "base": "<p>License Manager cannot allocate a license to a resource because of its state. </p> <p>For example,
you cannot allocate a license to an instance in the process of shutting down.</p>",
     "refs": {
     }
    },
    "InventoryFilter": {

   "base": "<p>An inventory filter object.</p>",
     "refs": {
       "InventoryFilterList$member": null
     }
    },
    "InventoryFilterCondition": {
     "base": null,
     "refs": {
       "InventoryFilter$Condition": "<p>The condition of the filter.</p>"
     }
    },
    "InventoryFilterList": {
     "base": null,
     "refs": {
       "ListResourceInventoryRequest$Filters": "<p>One or more filters.</p>"
     }
    },
    "LicenseConfiguration": {
     "base": "<p>A license configuration is an abstraction of a customer license agreement that can be consumed and
```

enforced by License Manager. Components include specifications for the license type (licensing by instance, socket, CPU, or VCPU), tenancy (shared tenancy, Amazon EC2 Dedicated Instance, Amazon EC2 Dedicated Host, or any of these), host affinity (how long a VM must be associated with a host), the number of licenses purchased and used.</p>",
      "refs": {
        "LicenseConfigurations$member": null
      }

    },
    "LicenseConfigurationAssociation": {
      "base": "<p>Describes a server resource that is associated with a license configuration.</p>",
      "refs": {
        "LicenseConfigurationAssociations$member": null
      }
    },
    "LicenseConfigurationAssociations": {
      "base": null,
      "refs": {
        "ListAssociationsForLicenseConfigurationResponse$LicenseConfigurationAssociations": "<p>Lists association objects for the license configuration, each containing the association time, number of consumed licenses, resource ARN, resource ID, account ID that owns the resource, resource size, and resource type.</p>"
      }
    },
    "LicenseConfigurationStatus": {
      "base": null,
      "refs": {
        "UpdateLicenseConfigurationRequest$LicenseConfigurationStatus": "<p>New status of the license configuration (<code>ACTIVE</code> or <code>INACTIVE</code>).</p>"
      }
    },
    "LicenseConfigurationUsage": {
      "base": "<p>Contains details of the usage of each resource
 from the license pool.</p>",
      "refs": {
        "LicenseConfigurationUsageList$member": null
      }
    },
    "LicenseConfigurationUsageList": {
      "base": null,
      "refs": {
        "ListUsageForLicenseConfigurationResponse$LicenseConfigurationUsageList": "<p>An array of <code>LicenseConfigurationUsage</code> objects.</p>"
      }
    },
    "LicenseConfigurations": {
      "base": null,
      "refs": {
        "ListLicenseConfigurationsResponse$LicenseConfigurations": "<p>Array of license configuration

objects.</p>"
    }
  },
  "LicenseCountingType": {
    "base": null,
    "refs": {
      "CreateLicenseConfigurationRequest$LicenseCountingType": "<p>Dimension to use to track the license inventory.</p>",
      "GetLicenseConfigurationResponse$LicenseCountingType": "<p>Dimension on which the licenses are counted (for example, instances, cores, sockets, or VCPUs).</p>",
      "LicenseConfiguration$LicenseCountingType": "<p>Dimension to use to track license inventory.</p>"
    }
  },
  "LicenseSpecification": {
    "base": "<p>Object used for associating a license configuration with a resource.</p>",
    "refs": {
      "LicenseSpecifications$member": null
    }
  },
  "LicenseSpecifications": {
    "base": null,
    "refs": {
      "ListLicenseSpecificationsForResourceResponse$LicenseSpecifications": "<p>License configurations associated with a resource.</p>",
      "UpdateLicenseSpecificationsForResourceRequest$AddLicenseSpecifications": "<p>License configuration ARNs to be added to a resource.</p>",
      "UpdateLicenseSpecificationsForResourceRequest$RemoveLicenseSpecifications": "<p>License configuration ARNs to be removed from a resource.</p>"
    }
  },
  "LicenseUsageException": {
    "base": "<p>You do not have enough licenses available to support a new resource launch.</p>",
    "refs": {
    }
  },
  "ListAssociationsForLicenseConfigurationRequest": {
    "base": null,
    "refs":
{
    }
  },
  "ListAssociationsForLicenseConfigurationResponse": {
    "base": null,
    "refs": {
    }
  },
  "ListLicenseConfigurationsRequest": {

```
    "base": null,
    "refs": {
    }
   },
   "ListLicenseConfigurationsResponse": {
    "base": null,
    "refs": {
    }
   },
   "ListLicenseSpecificationsForResourceRequest": {
    "base": null,
    "refs": {
    }
   },
   "ListLicenseSpecificationsForResourceResponse": {
    "base": null,
    "refs": {
    }
   },
   "ListResourceInventoryRequest": {
    "base": null,
    "refs": {
    }
   },
   "ListResourceInventoryResponse": {
    "base": null,
    "refs": {
    }
   },
   "ListTagsForResourceRequest": {
    "base": null,
    "refs": {
    }
   },
   "ListTagsForResourceResponse": {
    "base": null,
    "refs": {
    }
   },
   "ListUsageForLicenseConfigurationRequest": {
    "base": null,
    "refs": {
    }
   },
   "ListUsageForLicenseConfigurationResponse":
{
    "base": null,
    "refs": {
```

```
    }
  },
  "ManagedResourceSummary": {
   "base": "<p>Summary for a resource.</p>",
   "refs": {
     "ManagedResourceSummaryList$member": null
   }
  },
  "ManagedResourceSummaryList": {
   "base": null,
   "refs": {
     "GetLicenseConfigurationResponse$ManagedResourceSummaryList": "<p>List of summaries of managed
resources.</p>",
     "LicenseConfiguration$ManagedResourceSummaryList": "<p>List of summaries for managed resources.</p>"
   }
  },
  "Message": {
   "base": null,
   "refs": {
     "AccessDeniedException$Message": null,
     "AuthorizationException$Message": null,
     "FailedDependencyException$Message": null,
     "FilterLimitExceededException$Message": null,
     "InvalidParameterValueException$Message": null,
     "InvalidResourceStateException$Message": null,
     "LicenseUsageException$Message": null,
     "RateLimitExceededException$Message":
null,
     "ResourceLimitExceededException$Message": null,
     "ServerInternalException$Message": null
   }
  },
  "OrganizationConfiguration": {
   "base": "<p>Object containing configuration information for AWS Organizations.</p>",
   "refs": {
     "GetServiceSettingsResponse$OrganizationConfiguration": "<p>Indicates whether AWS Organizations has
been integrated with License Manager for cross-account discovery.</p>",
     "UpdateServiceSettingsRequest$OrganizationConfiguration": "<p>Integrates AWS Organizations with License
Manager for cross-account discovery.</p>"
   }
  },
  "RateLimitExceededException": {
   "base": "<p>Too many requests have been submitted. Try again after a brief wait.</p>",
   "refs": {
   }
  },
  "ResourceInventory": {
   "base": "<p>A set of attributes that describe a resource.</p>",
```

      "refs": {
        "ResourceInventoryList$member": null
      }
    },
    "ResourceInventoryList": {

   "base": null,
      "refs": {
        "ListResourceInventoryResponse$ResourceInventoryList": "<p>The detailed list of resources.</p>"
      }
    },
    "ResourceLimitExceededException": {
      "base": "<p>Your resource limits have been exceeded.</p>",
      "refs": {
      }
    },
    "ResourceType": {
      "base": null,
      "refs": {
        "ConsumedLicenseSummary$ResourceType": "<p>Resource type of the resource consuming a license
(instance, host, or AMI).</p>",
        "LicenseConfigurationAssociation$ResourceType": "<p>Type of server resource.</p>",
        "LicenseConfigurationUsage$ResourceType": "<p>Type of resource associated with athe license
configuration.</p>",
        "ManagedResourceSummary$ResourceType": "<p>Type of resource associated with a license (instance, host,
or AMI).</p>",
        "ResourceInventory$ResourceType": "<p>The type of resource.</p>"
      }
    },
    "ServerInternalException": {
      "base": "<p>The server experienced an internal error. Try
again.</p>",
      "refs": {
      }
    },
    "String": {
      "base": null,
      "refs": {
        "CreateLicenseConfigurationRequest$Name": "<p>Name of the license configuration.</p>",
        "CreateLicenseConfigurationRequest$Description": "<p>Human-friendly description of the license
configuration.</p>",
        "CreateLicenseConfigurationResponse$LicenseConfigurationArn": "<p>ARN of the license configuration
object after its creation.</p>",
        "DeleteLicenseConfigurationRequest$LicenseConfigurationArn": "<p>Unique ID of the configuration object to
delete.</p>",
        "GetLicenseConfigurationRequest$LicenseConfigurationArn": "<p>ARN of the license configuration being
requested.</p>",
        "GetLicenseConfigurationResponse$LicenseConfigurationId": "<p>Unique ID for the license

configuration.</p>",
　　"GetLicenseConfigurationResponse$LicenseConfigurationArn": "<p>ARN of the license configuration requested.</p>",
　　"GetLicenseConfigurationResponse$Name": "<p>Name of the license configuration.</p>",
　　"GetLicenseConfigurationResponse$Description": "<p>Description of the license configuration.</p>",
　　"GetLicenseConfigurationResponse$Status": "<p>License configuration status (active, etc.).</p>",
　　"GetLicenseConfigurationResponse$OwnerAccountId": "<p>Owner account ID for the license configuration.</p>",
　　"GetServiceSettingsResponse$S3BucketArn": "<p>Regional S3 bucket path for storing reports, license trail event data, discovery data, etc.</p>",
　　"GetServiceSettingsResponse$SnsTopicArn": "<p>SNS topic configured to receive notifications from License Manager.</p>",
　　"InventoryFilter$Name": "<p>The name of the filter.</p>",
　　"InventoryFilter$Value": "<p>Value of the filter.</p>",
　　"LicenseConfiguration$LicenseConfigurationId": "<p>Unique ID of the <code>LicenseConfiguration</code> object.</p>",
　　"LicenseConfiguration$LicenseConfigurationArn": "<p>ARN of the <code>LicenseConfiguration</code> object.</p>",
　　"LicenseConfiguration$Name": "<p>Name of the license configuration.</p>",
　　"LicenseConfiguration$Description": "<p>Description of the license configuration.</p>",
　　"LicenseConfiguration$Status": "<p>Status of the license configuration.</p>",
　　"LicenseConfiguration$OwnerAccountId": "<p>Account ID of the license configuration's owner.</p>",
　　"LicenseConfigurationAssociation$ResourceArn": "<p>ARN of the resource associated with the license configuration.</p>",
　　"LicenseConfigurationAssociation$ResourceOwnerId": "<p>ID of the AWS account that owns the resource consuming licenses.</p>",
　　"LicenseConfigurationUsage$ResourceArn": "<p>ARN of the resource associated with a license configuration.</p>",
　　"LicenseConfigurationUsage$ResourceStatus": "<p>Status of a resource associated with the license configuration.</p>",
　　"LicenseConfigurationUsage$ResourceOwnerId": "<p>ID of the account that owns a resource that is associated with the license configuration.</p>",
　　"LicenseSpecification$LicenseConfigurationArn": "<p>ARN of the <code>LicenseConfiguration</code> object.</p>",
　　"ListAssociationsForLicenseConfigurationRequest$LicenseConfigurationArn": "<p>ARN of a <code>LicenseConfiguration</code> object.</p>",
　　"ListAssociationsForLicenseConfigurationRequest$NextToken": "<p>Token for the next set of results.</p>",
　　"ListAssociationsForLicenseConfigurationResponse$NextToken": "<p>Token for the next set of results.</p>",
　　"ListLicenseConfigurationsRequest$NextToken": "<p>Token for the next set of results.</p>",
　　"ListLicenseConfigurationsResponse$NextToken": "<p>Token for the next set of results.</p>",
　　"ListLicenseSpecificationsForResourceRequest$ResourceArn": "<p>ARN of an AMI or Amazon EC2 instance that has an associated license configuration.</p>",
　　"ListLicenseSpecificationsForResourceRequest$NextToken": "<p>Token for the next set of results.</p>",
　　"ListLicenseSpecificationsForResourceResponse$NextToken": "<p>Token for the next set of results.</p>",
　　"ListResourceInventoryRequest$NextToken": "<p>Token for the next set of results.</p>",
　　"ListResourceInventoryResponse$NextToken": "<p>Token for the next set of results.</p>",
　　"ListTagsForResourceRequest$ResourceArn": "<p>ARN for the resource.</p>",

"ListUsageForLicenseConfigurationRequest$LicenseConfigurationArn": "<p>ARN of the targeted <code>LicenseConfiguration</code> object.</p>",
    "ListUsageForLicenseConfigurationRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListUsageForLicenseConfigurationResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ResourceInventory$ResourceId": "<p>Unique ID of the resource.</p>",
    "ResourceInventory$ResourceArn": "<p>The ARN of the resource.</p>",
    "ResourceInventory$Platform": "<p>The platform of the resource.</p>",
    "ResourceInventory$PlatformVersion":
 "<p>Platform version of the resource in the inventory.</p>",
    "ResourceInventory$ResourceOwningAccountId": "<p>Unique ID of the account that owns the resource.</p>",
    "StringList$member": null,
    "Tag$Key": "<p>Key for the resource tag.</p>",
    "Tag$Value": "<p>Value for the resource tag.</p>",
    "TagKeyList$member": null,
    "TagResourceRequest$ResourceArn": "<p>Resource of the ARN to be tagged.</p>",
    "UntagResourceRequest$ResourceArn": "<p>ARN of the resource.</p>",
    "UpdateLicenseConfigurationRequest$LicenseConfigurationArn": "<p>ARN for a license configuration.</p>",
    "UpdateLicenseConfigurationRequest$Name": "<p>New name of the license configuration.</p>",
    "UpdateLicenseConfigurationRequest$Description": "<p>New human-friendly description of the license configuration.</p>",
    "UpdateLicenseSpecificationsForResourceRequest$ResourceArn": "<p>ARN for an AWS server resource.</p>",

    "UpdateServiceSettingsRequest$S3BucketArn": "<p>ARN of the Amazon S3 bucket where License Manager information is stored.</p>",
    "UpdateServiceSettingsRequest$SnsTopicArn": "<p>ARN of the Amazon SNS topic used for License Manager alerts.</p>"
   }
  },
  "StringList": {
   "base": null,
   "refs": {
    "CreateLicenseConfigurationRequest$LicenseRules": "<p>Array of configured License Manager rules.</p>",
    "GetLicenseConfigurationResponse$LicenseRules": "<p>List of flexible text strings designating license rules.</p>",
    "LicenseConfiguration$LicenseRules": "<p>Array of configured License Manager rules.</p>",
    "ListLicenseConfigurationsRequest$LicenseConfigurationArns": "<p>An array of ARNs for the calling accounts license configurations.</p>",
    "UpdateLicenseConfigurationRequest$LicenseRules": "<p>List of flexible text strings designating license rules.</p>"
   }
  },
  "Tag": {
   "base": "<p>Tag for a resource in a key-value
 format.</p>",
   "refs": {
    "TagList$member": null
   }

```
    },
    "TagKeyList": {
      "base": null,
      "refs": {
        "UntagResourceRequest$TagKeys": "<p>List keys identifying tags to remove.</p>"
      }
    },
    "TagList": {
      "base": null,
      "refs": {
        "CreateLicenseConfigurationRequest$Tags": "<p>The tags to apply to the resources during launch. You can
only tag instances and volumes on launch. The specified tags are applied to all instances or volumes that are created
during launch. To tag a resource after it has been created, see CreateTags .</p> <p/>",
        "GetLicenseConfigurationResponse$Tags": "<p>List of tags attached to the license configuration.</p>",
        "ListTagsForResourceResponse$Tags": "<p>List of tags attached to the resource.</p>",
        "TagResourceRequest$Tags": "<p>Names of the tags to attach to the resource.</p>"
      }
    },
    "TagResourceRequest": {
      "base": null,
      "refs": {
      }
    },

    "TagResourceResponse": {
      "base": null,
      "refs": {
      }
    },
    "UntagResourceRequest": {
      "base": null,
      "refs": {
      }
    },
    "UntagResourceResponse": {
      "base": null,
      "refs": {
      }
    },
    "UpdateLicenseConfigurationRequest": {
      "base": null,
      "refs": {
      }
    },
    "UpdateLicenseConfigurationResponse": {
      "base": null,
      "refs": {
      }
```

```
    },
    "UpdateLicenseSpecificationsForResourceRequest": {
      "base": null,
      "refs": {
      }
    },
    "UpdateLicenseSpecificationsForResourceResponse": {
      "base": null,
      "refs": {
      }
    },
    "UpdateServiceSettingsRequest": {
      "base": null,
      "refs": {
      }
    },
    "UpdateServiceSettingsResponse": {
      "base": null,
      "refs": {
      }
    }
  }
}
```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2015 James Saryerwinnie

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

<div align="center">
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
</div>

 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

 1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership
 of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

 4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
{
 "pagination": {
 }
}

1.31 aws-aws-sdk-go-v2-internal-configsources 1.2.7

1.31.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```go
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil,
 err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}
```

```go
type AssociateLicenseOutput struct {

	// A structure containing data about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
	if err !=
	nil {
		return err
	}
	err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
```

```
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack);
    err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
    middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack,
    options); err != nil {
    return err
    }
    return nil
    }
```

```
func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

```
name: License Scan

on: [pull_request]

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
    matrix:
      python-version: [3.9]

   steps:
    - name: Checkout target
      uses: actions/checkout@v2
      with:
       path: sdkbase
       ref: ${{ github.base_ref }}
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
       path: new-ref
       fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
       repository: nexB/scancode-toolkit
       path: scancode-toolkit
       fetch-depth: 1
    - name: Set up Python ${{ matrix.python-version }}
```

```yaml
      uses: actions/setup-python@v2
      with:
       python-version: ${{ matrix.python-version }}
     # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)


// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }


 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }


 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```go
type CheckInLicenseInput struct {

	// License consumption token.
	//
	// This member is required.
	LicenseConsumptionToken *string

	// License beneficiary.
	Beneficiary
	*string

	noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
```

```
    return err
  }
  if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
  }
  if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
  }
  if err = addRetryMiddlewares(stack, options); err != nil {
    return err
  }
  if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
  }
  if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
  }
  if err = addClientUserAgent(stack, options); err != nil {
    return err
  }
  if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
  }
  if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
  }
  if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
  }
  if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return
err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
    return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
    return err
```

```
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
    Region:        region,
    ServiceID:     ServiceID,
    OperationName: "CheckInLicense",
    }
}
```

```
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
            "type": "structure",
            "members": {
                "Name": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
                    }
                },
                "Values": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                    "traits": {
                        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
                    }
                },
                "Operator": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
                    "traits": {
                        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
```

```
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
               "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
         },
         "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
               "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n      Organizations integration has been enabled, and
 which Regions data will be aggregated from.</p>"
            }
         },
         "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
               "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
         },
         "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
               "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
         },
         "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays
 the aggregated data for Linux\n      subscriptions.</p>"
            }
         }
      }
   },
   "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
         "AmiId": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
         },
         "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
```

```
                        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
                      }
                  },
                  "InstanceType": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
                      }

                  },
                  "AccountID": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
                      }
                  },
                  "Status": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The status of the instance.</p>"
                      }
                  },
                  "Region": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
                      }
                  },
                  "UsageOperation": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
                      }
                  },
                  "ProductCode": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
                    "traits": {
                        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
                      }
                  },
                  "LastUpdatedTime": {
                    "target": "smithy.api#String",
                    "traits": {
```

```
              "smithy.api#documentation": "<p>The time in which the last discovery
updated the instance details.</p>"
                }
            },
            "SubscriptionName": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
          "type": "service",
          "version": "2018-05-10",
          "operations": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
            },
            {
```

```
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
                    }
            ],

        "traits": {
            "aws.api#service": {
                "sdkId": "License Manager Linux Subscriptions",
                "arnNamespace": "license-manager-linux-subscriptions",
                "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
            },
            "aws.auth#sigv4": {
                "name": "license-manager-linux-subscriptions"
            },
            "aws.protocols#restJson1": {},
            "smithy.api#cors": {
                "additionalAllowedHeaders": [
                    "*",
                    "content-type",
                    "x-amz-content-sha256",
                    "x-amz-user-agent",
                    "x-amzn-platform-id",
                    "x-amzn-trace-id",
                    "content-length",
                    "x-api-key",
                    "authorization",
                    "x-amz-date",

    "x-amz-security-token",
                    "Access-Control-Allow-Headers",
                    "Access-Control-Allow-Methods",
                    "Access-Control-Allow-Origin"
                ],
                "additionalExposedHeaders": [
                    "x-amzn-errortype",
                    "x-amzn-requestid",
                    "x-amzn-trace-id"
                ]
            },
            "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager Linux Subscriptions",
            "smithy.rules#endpointRuleSet": {
                "version": "1.0",
                "parameters": {
                    "Region": {
                        "builtIn": "AWS::Region",
                        "required": false,
                        "documentation": "The
```

AWS region used to dispatch the request.",
                    "type": "String"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

                    "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }
            },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {

                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {

```
                "ref": "UseFIPS"
              },
                true
            ]
          }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [],
        "type": "tree",
        "rules": [
          {

        "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
              "type": "error"
          },
          {
            "conditions": [],

    "endpoint": {
                "url": {
                  "ref": "Endpoint"
                },
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
          }
        ]
      }
    ]
  },
  {
    "conditions": [],
    "type": "tree",
```

```
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",

                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            }
        ],
        "type": "tree",
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "aws.partition",
                        "argv": [
                            {
                                "ref": "Region"
                            }
                        ],

                        "assign": "PartitionResult"
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            },

                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
```

```json
                                        true
                                    ]
                                }
                            ],
                            "type": "tree",
                            "rules": [
                                {
                                    "conditions": [
                                        {

                                        "fn": "booleanEquals",
                                            "argv": [
                                                true,
                                                {
                                                    "fn": "getAttr",
                                                    "argv": [
                                                        {
                                                            "ref": "PartitionResult"
                                                        },
                                                        "supportsFIPS"
                                                    ]

            }

                                    ]
                                },
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },

                                    "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {
                                "conditions": [],
                                "type": "tree",
                                "rules": [
```

```
                                    {
            "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                              },
                            "type": "endpoint"
                          }
                        ]
                      }
                    ]

                },
                    {
                      "conditions": [],
                      "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

                      "type": "error"
                    }
                  ]
                },
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {

              "ref": "UseFIPS"
                        },
                        true
                      ]
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,

          {

                              "fn": "getAttr",
```

```
                                          "argv": [
                                              {
                                                  "ref": "PartitionResult"
                                              },
                                              "supportsFIPS"
                                          ]
                                      }
                                  ]
                              }
                          ],
                          "type": "tree",

                          "rules": [
                              {
                                  "conditions": [],
                                  "type": "tree",
                                  "rules": [
                                      {
                                          "conditions": [],
                                          "endpoint": {
                                              "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                                              "properties": {},
                                              "headers": {}

                                          },
                                          "type": "endpoint"
                                      }
                                  ]
                              }
                          ]
                      },
                      {
                          "conditions": [],
                          "error": "FIPS is enabled but this partition does not support FIPS",
                          "type": "error"
                      }
                  ]

              },
                      {
                          "conditions": [
                              {
                                  "fn": "booleanEquals",
                                  "argv": [
                                      {
                                          "ref": "UseDualStack"
                                      },
```

```
                                        true
                                    ]
                                }
                            ],
                            "type": "tree",
                            "rules": [
                                {

                                    "conditions": [
                                        {
                                            "fn": "booleanEquals",
                                            "argv": [
                                                true,
                                                {
                                                    "fn": "getAttr",
                                                    "argv": [
                                                        {
                                                            "ref": "PartitionResult"
                                                        },
                                                        "supportsDualStack"
                                                    ]
                                                }
                                            ]
                                        }
                                    ],
                                    "type": "tree",
                                    "rules": [
                                        {
                                            "conditions": [],
                                            "type": "tree",
                                            "rules": [
                                                {
                                                    "conditions": [],

                                            "endpoint": {
                                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                        "properties": {},
                                                        "headers": {}
                                                    },
                                                    "type": "endpoint"
                                                }
                                            ]
                                        }
                                    ]
                                },
```

```
                    {
                                    "conditions": [],
                                    "error": "DualStack is enabled but this partition does not support DualStack",
                                    "type": "error"
                          }
                       ]
                    },
                    {
                       "conditions": [],
                       "type": "tree",
                       "rules": [
                          {
                             "conditions": [],
                             "endpoint": {
                                "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                             },
                             "type": "endpoint"
                          }
                       ]
                    }
                 ]
              }
           ]
        },
        {
           "conditions": [],
           "error": "Invalid Configuration: Missing Region",
           "type": "error"
        }
     ]
   }
 ]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
```

```
              "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {

"endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
```

```
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
         "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint":

{

          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
```

```
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
             "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint":
{

              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
```

```
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {

        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
          }
```

```
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {

           "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
```

```
                    "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                 }
              },
              "params": {
                 "Region": "us-isob-east-1",

   "UseFIPS": false,
                 "UseDualStack": false
              }
           },
           {
              "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
              "expect": {
                 "endpoint": {
                    "url": "https://example.com"
                 }
              },
              "params": {
                 "Region": "us-east-1",
                 "UseFIPS": false,
                 "UseDualStack": false,
                 "Endpoint": "https://example.com"
              }
           },
           {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",

      "expect": {
                 "endpoint": {
                    "url": "https://example.com"
                 }
              },
              "params": {
                 "UseFIPS": false,
                 "UseDualStack": false,
                 "Endpoint": "https://example.com"
              }
           },
           {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {
                 "Region": "us-east-1",
                 "UseFIPS": true,
                 "UseDualStack": false,
```

```
                            "Endpoint": "https://example.com"
                        }
                    },
                    {
                        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                        "expect": {
                            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                        },
                        "params": {
                            "Region": "us-east-1",
                            "UseFIPS": false,
                            "UseDualStack": true,
                            "Endpoint": "https://example.com"
                        }
                    },
                    {
                        "documentation": "Missing region",
                        "expect": {
                            "error": "Invalid Configuration: Missing Region"
                        }

                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
```

```
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
         "traits": {
           "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
             "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type":
"operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with
commercial Linux\n      subscriptions.</p>",
      "smithy.api#http": {

  "code": 200,
        "method": "POST",
```

```
            "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by
 the name of\n      <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n      <p>The valid names for this filter
are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>AmiID</code>\n          </p>\n
</li>\n        <li>\n          <p>\n            <code>InstanceID</code>\n          </p>\n        </li>\n
<li>\n          <p>\n            <code>AccountID</code>\n          </p>\n        </li>\n        <li>\n
<p>\n            <code>Status</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>Region</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>UsageOperation</code>\n          </p>\n        </li>\n        <li>\n
   <p>\n            <code>ProductCode</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
   <code>InstanceType</code>\n          </p>\n        </li>\n      </ul>\n      <p>The valid Operators for this
filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>contains</code>\n          </p>\n
</li>\n        <li>\n          <p>\n            <code>equals</code>\n          </p>\n        </li>\n        <li>\n
     <p>\n            <code>Notequal</code>\n          </p>\n        </li>\n      </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

  }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
```

```
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
        "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
                "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
```

linked your\n   organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
            }

          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
```
that match one or\n   more sets of key-value pairs that you specify. For example, you can filter by the name of\n <code>Subscription</code> with an optional operator to see subscriptions that match,\n   partially match, or don't match a certain subscription's name.</p>\n   <p>The valid names for this filter are:</p>\n   \n \n   <p>\n   <code>Subscription</code>\n   </p>\n   \n   \n <p>The valid Operators for this filter are:</p>\n

     \n   \n   <p>\n   <code>contains</code>\n   </p>\n   \n \n   <p>\n   <code>equals</code>\n   </p>\n   \n   \n   <p>\n <code>Notequal</code>\n   </p>\n   \n   "

```
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                "smithy.api#length": {

                  "min": 1,
                    "max": 16384
```

```
                    }
                  }
                }
              },
              "traits": {
                "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
              }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
              "type": "structure",
              "members": {
                "Subscriptions": {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                  "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                  }
                },
                "NextToken": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                  }
                }
              }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
              "type": "string",
              "traits": {
                "smithy.api#enum": [
                  {
                    "name": "EQUAL",
                    "value": "Equal",
                    "documentation": "Equal operator"
                  },
                  {
                    "name": "NOT_EQUAL",
                    "value": "NotEqual",
                    "documentation": "Not equal operator"
                  },
                  {
                    "name": "CONTAINS",
                    "value": "Contains",
                    "documentation": "Contains operator"
                  }
                ],
                "smithy.api#length": {
```

```
            "min": 1,
            "max": 20
          }
        }
    },

    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled OrganizationIntegration"
            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled OrganizationIntegration"
            }
          ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [

{
            "name": "InProgress",
            "value": "InProgress",
            "documentation": "InProgress status"
          },
          {
            "name": "Completed",
            "value": "Completed",
            "documentation": "Completed status"
          },
          {
            "name": "Successful",
            "value": "Successful",
            "documentation": "Successful status"
```

```
              },
              {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#length": {
                "min": 1,
                "max": 100
              }
            }
          },
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 100
            },
            "smithy.api#uniqueItems": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
          "type": "map",
          "key": {
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"

          }
            },
```

```
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },

    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
```

```
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",

  "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
```

```
                }
              }
            }
          },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },

          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
```

```
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
      }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //   - Beneficiary
 //   - ProductSKU
```

```go
	//   - Fingerprint
	//
	//   - Status
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListLicensesOutput struct {

	// License details.
	Licenses []types.License

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	if
	err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}
```

```go
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
	return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
	return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
	return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
	return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
	return err
}
if err = addClientUserAgent(stack, options); err != nil {
	return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
	return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
	return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
	return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
	return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
```

```go
    return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
    return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
    return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
  }
  return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
  return &awsmiddleware.RegisterServiceMetadata{
    Region:       region,
    ServiceID:
      ServiceID,
    OperationName: "ListLicenses",
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
  "context"
  "fmt"
  awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
  "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
  "github.com/aws/aws-sdk-go-v2/service/grafana/types"
  "github.com/aws/smithy-go/middleware"
  smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
  if params == nil {
    params = &DisassociateLicenseInput{}
  }

  result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
  if err != nil {
    return nil, err
  }
```

```go
    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    //
    This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
```

```go
if err := addProtocolFinalizerMiddlewares(stack,
options, "DisassociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
```

```go
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
  }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
  }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
  }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
  }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
  }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
  }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
 {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
```

```go
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type
 CheckoutLicenseInput struct {

 // Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string
```

```go
	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

	// Checkout type.
	CheckoutType types.CheckoutType

	// Allowed license entitlements.
	EntitlementsAllowed []types.EntitlementData

	// Date and time at which the license checkout expires.
	Expiration *string

	// Date and time at which the license checkout is issued.
	IssuedAt *string

	// Amazon Resource
	Name (ARN) of the checkout license.
	LicenseArn *string

	// License consumption token.
	LicenseConsumptionToken *string

	// Node ID.
	NodeId *string

	// Signed token.
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
```

```go
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil
{
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack);
err != nil {
 return err
```

```go
	}
	if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```go
// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier
 that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see RFC 7518 Digital Signature with
 // RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata
```

```go
 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```go
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return
err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
```

```go
	}
	if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutBorrowLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```go
// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide
 to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string
```

```
// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity
*types.DatetimeRange

// Information about the license.
LicenseMetadata []types.Metadata

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
```

```go
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
return fmt.Errorf("add
protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
```

```go
}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
	err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName:
"CreateLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```go
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version
 *string

 noSmithyDocumentSerde
}

type GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
```

```go
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return
err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
```

```go
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpGetLicenseValidationMiddleware(stack);
    err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "GetLicense",
 }
}
```
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns ...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
```

```go
    c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
     return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
      - ProductSKU
    //   - Status
    //   - Fingerprint
    //   - IssuerName
    //   - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
```

```go
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack);
err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
```

```go
  if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
  }
  if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
  }
  if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
  return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
  return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
  }
  return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "ListReceivedLicenses",
 }
}
```

                    Apache License
                Version 2.0, January 2004
              http://www.apache.org/licenses/

 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct,
indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
    incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```json
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Details about an Active Directory identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
```

```
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
         },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n        <note>\n           <p>Your estimated bill for charges on the number of users and related
costs will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
     monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/AssociateUser"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n      resource.</p>",
          "smithy.api#error": "server"
```

```
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
```

```
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
```

```
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/user/DisassociateUser"
                },
                "smithy.api#idempotent": {}
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
            "type": "structure",
            "members": {
                "Username": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                        "smithy.api#required": {}
                    }
                },
                "InstanceId": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                        "smithy.api#required": {}
                    }
                },
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
                        "smithy.api#required": {}
```

```
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },

"com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used
```

to return more specific results from a describe\n   operation. Filters can be used to match a set of resources by specific criteria, such as tags,\n   attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>An object that details the registered identity providers product related configuration\n   settings such as the subnets to provision VPC endpoints.</p>",
                "smithy.api#required":
{}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",

```
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status
of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
```

```json
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
```

```
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
```

```
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",

      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },
    },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
         },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"

   ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n     a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
```

```
"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "String"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack
endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
  "type": "Boolean"
},
"UseFIPS": {
  "builtIn": "AWS::UseFIPS",
  "required": true,
  "default": false,
  "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
  "type": "Boolean"
},
"Endpoint": {
  "builtIn": "SDK::Endpoint",
  "required": false,
  "documentation": "Override the endpoint used to send this request",
  "type": "String"
}

},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ],
    "type": "tree",
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
```

```
                              "argv": [
                                {
                                  "ref": "UseFIPS"
                                },

                                true
                              ]
                            }
                          ],
                          "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                          "type": "error"
                        },
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                {
                                  "ref": "UseDualStack"
                                },
                                true
                              ]
                            }
                          ],

      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                          "type": "error"
                        },
                        {
                          "conditions": [],
                          "endpoint": {
                            "url": {
                              "ref": "Endpoint"
                            },
                            "properties": {},
                            "headers": {}
                          },
                          "type": "endpoint"
                        }
                      ]
                    },
                    {
                      "conditions": [
                        {
                          "fn": "isSet",
                          "argv": [
```

```
                {
                                "ref": "Region"
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "type": "tree",

"rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },
                                        true
                                    ]
                                },
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"

                                        },
                                        true
                                    ]
                                }
                            ],
                            "type": "tree",
                            "rules": [
                                {
```

```
"conditions": [
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsFIPS"
        ]
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"type": "tree",
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
]
```

```
            },
            {
              "conditions": [],
              "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
              "type": "error"
            }
          ]
        },
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseFIPS"
                },
                true
              ]
            }
          ],
          "type": "tree",

    "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsFIPS"

                      ]
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [],
                  "endpoint": {
```

```
                                    "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}

                    },
                                    "type": "endpoint"
                                }
                            ]
                        },
                        {
                            "conditions": [],
                            "error": "FIPS is enabled but this partition does not support FIPS",
                            "type": "error"
                        }
                    ]
                },
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {

                                        "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                            ]
                                        }
                                    ]
```

```
                                          }
                                      ],
                                      "type": "tree",
                                      "rules": [
                                          {

                                              "conditions": [],
                                              "endpoint": {
                                                  "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                  "properties": {},
                                                  "headers": {}
                                              },
                                              "type": "endpoint"
                                          }
                                      ]
                                  },
                                  {
                                      "conditions": [],
                                      "error": "DualStack is enabled but this partition does
not support DualStack",
                                      "type": "error"
                                  }
                              ]
                          },
                          {
                              "conditions": [],
                              "endpoint": {
                                  "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                              },
                              "type": "endpoint"
                          }
                      ]
                  }
              ]
          },
          {
              "conditions":
[],
              "error": "Invalid Configuration: Missing Region",
              "type": "error"
          }
      ]
  },
  "smithy.rules#endpointTests": {
```

```
"testCases": [
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
      }
    },

    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
      }
```

```
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with
 FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },

     "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
```

```
            },
            {
                "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },

        "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
```

```
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
              }
            },

      "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
            "expect":
{
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
```

```
              "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "error": "DualStack is enabled but this partition does not support DualStack"
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": false,
                "UseDualStack":
true
              }
            },
            {
              "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                }
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
              },
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                }
              },
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
```

```json
        },
        {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                }
            },
            "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {

"documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
```

```
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "Missing region",
          "expect": {
            "error": "Invalid Configuration: Missing Region"
          }
        }
      ],
      "version": "1.0"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated":
{

            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
```

```
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify.</p>"
          }
        }
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type":
"structure",
        "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
```

```
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListProductSubscriptions"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
```

```
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n    more sets of key-value pairs that you specify.</p>"
              }
            },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
          },

      "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target":
```

"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
          },
          "smithy.api#paginated":
{

            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
              "smithy.api#required": {}

```
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {

        "Username": {
          "target": "smithy.api#String",
          "traits": {
```

```
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }

    },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for a user.</p>",
              "smithy.api#required": {}
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
```

```
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/RegisterIdentityProvider"
          },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n      subnets to provision VPC endpoints.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
```

```
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",

 "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      },
      "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
```

```
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n      your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n      subnets to provision VPC endpoints, and the security group ID that
 is associated with the VPC\n      endpoints. The security group should permit inbound TCP port 1688
communication from resources\n      in the VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n        <note>\n          <p>Your estimated bill for charges on the number of users and related costs
will take 48\n         hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
 For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n        monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StartProductSubscription"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
```

```
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"


        },
        {

            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
```

```
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required":
{}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation":
```

"<p>Metadata that describes the start product subscription operation.</p>",

        "smithy.api#required": {}

      }

    }

  }
},
"com.amazonaws.licensemanagerusersubscriptions#StringList": {

  "type": "list",

  "member": {

    "target": "smithy.api#String"

  }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {

  "type": "string",

  "traits": {

    "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"

  }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {

  "type": "list",

  "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"

  }
},
"com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {

  "type": "structure",

  "members": {

    "message": {

     "target": "smithy.api#String"

    }

  },

  "traits": {

    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",

    "smithy.api#error": "client"

  }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {

  "type": "operation",

  "input": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"

  },

  "output": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"

  },

  "errors": [

    {

```
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                 },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity\n     provider.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/identity-provider/UpdateIdentityProviderSettings"
                },
                "smithy.api#idempotent": {}
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
            "type": "structure",
            "members": {
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#required": {}
                    }
                },
                "Product": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                        "smithy.api#required": {}
                    }
                },
                "UpdateSettings": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                    "traits": {
                        "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n     update any combination of settings in a single operation such as the:</p>\n
<ul>\n        <li>\n            <p>Subnets which you want to add to provision VPC endpoints.</p>\n        </li>\n
       <li>\n            <p>Subnets which you want to remove the VPC endpoints from.</p>\n        </li>\n
<li>\n            <p>Security group ID which permits traffic to the VPC endpoints.</p>\n        </li>\n     </ul>",
                        "smithy.api#required": {}
```

```
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {

            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n     require connectivity to activation servers.</p>",
              "smithy.api#required": {}
            }
          },
          "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID
 that allows inbound TCP port 1688 communication between resources in\n     your VPC and the VPC endpoints for
activation servers.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as\n     the subnets to provision VPC endpoints.</p>"
        }
      },
```

```json
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
  }
  {
    "smithy": "2.0",
    "metadata": {
      "suppressions": [
        {
          "id": "HttpMethodSemantics",
          "namespace": "*"
        },
        {
          "id": "HttpResponseCodeSemantics",
          "namespace": "*"
        },
        {
          "id": "PaginatedTrait",
          "namespace": "*"
        },
        {
          "id": "HttpHeaderTrait",
          "namespace": "*"
        },
        {
          "id": "HttpUriConflict",
          "namespace": "*"
        },
        {
          "id": "Service",
          "namespace": "*"
        }
      ]
    },
    "shapes": {
      "com.amazonaws.licensemanager#AWSLicenseManager": {
        "type": "service",
        "version": "2018-08-01",
```

```
"operations": [
    {
        "target": "com.amazonaws.licensemanager#AcceptGrant"
    },

{

        "target": "com.amazonaws.licensemanager#CheckInLicense"
    },
    {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
    },
    {
        "target": "com.amazonaws.licensemanager#CheckoutLicense"
    },
    {
        "target": "com.amazonaws.licensemanager#CreateGrant"
    },
    {
        "target": "com.amazonaws.licensemanager#CreateGrantVersion"
    },
    {
        "target": "com.amazonaws.licensemanager#CreateLicense"
    },
    {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
    },
    {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
    },
    {
        "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
     },
    {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
    },
    {
        "target": "com.amazonaws.licensemanager#CreateToken"
    },
    {
        "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
        "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
```

      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
      "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    },

    },
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
    {
      "target": "com.amazonaws.licensemanager#GetGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseUsage"
    },
    {
      "target": "com.amazonaws.licensemanager#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
     },
    {
      "target": "com.amazonaws.licensemanager#ListDistributedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
    },

```
{
    "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenses"
},
{
    "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
},
{

    "target": "com.amazonaws.licensemanager#ListLicenseVersions"
},
{
    "target": "com.amazonaws.licensemanager#ListReceivedGrants"
},
{
    "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
},
{
    "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
},
{
    "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
},
{
    "target": "com.amazonaws.licensemanager#ListResourceInventory"
},
{
    "target": "com.amazonaws.licensemanager#ListTagsForResource"
},
{
    "target": "com.amazonaws.licensemanager#ListTokens"
},
{
    "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
},
{
    "target": "com.amazonaws.licensemanager#RejectGrant"
},
{
    "target": "com.amazonaws.licensemanager#TagResource"
},
{
    "target": "com.amazonaws.licensemanager#UntagResource"
},
{
    "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
}
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager",
        "arnNamespace": "license-manager",
        "cloudFormationName": "LicenseManager",
        "cloudTrailEventSource": "licensemanager.amazonaws.com",
        "endpointPrefix": "license-manager"
      },
      "aws.auth#sigv4": {
        "name": "license-manager"
      },
      "aws.protocols#awsJson1_1": {},
      "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n       Amazon Web Services accounts and on-premises servers.</p>",
      "smithy.api#title": "AWS License Manager",
      "smithy.api#xmlNamespace": {
        "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
      },
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
          "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
          "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
```

```
                    "required": true,
                    "default": false,
                    "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
                    "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }
            },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]

                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },
                                        true
                                    ]
                                }
                            ],
                            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                            "type": "error"
                        },
                        {

     "conditions": [
                                {
```

```
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ]
          },
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "type": "tree",
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "aws.partition",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ],
                    "assign": "PartitionResult"
```

```
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref":
"UseFIPS"
                        },
                        true
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                              ]

                            }
                          ]
                        },
                        {
                          "fn": "booleanEquals",
```

```
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },

    "supportsDualStack"

                        ]
                      }
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                      "properties": {},
                      "headers": {}

                  },
                    "type": "endpoint"
                  }
                ]
              },
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",

                "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
```

"error": "FIPS and DualStack are enabled, but this partition does not support one or both",

"type": "error"

```
                                 ]
                               }
                             ],
                          "type": "tree",
                          "rules": [
                             {
                                "conditions": [
                                   {
                                      "fn": "booleanEquals",
                                      "argv": [
                                         true,

                                   {
                                            "fn": "getAttr",
                                            "argv": [
                                               {
                                                  "ref": "PartitionResult"
                                               },
                                               "supportsFIPS"
                                            ]
                                         }
                                      ]
                                   }
                                ],
                                "type": "tree",
                                "rules": [

                                   {
                                         "conditions": [],
                                         "endpoint": {
                                            "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                         },
                                         "type": "endpoint"
                                      }
                                   ]
                                },
                                {
                                   "conditions": [],
                                   "error": "FIPS is enabled but this
partition does not support FIPS",
                                   "type": "error"
                                }
                             ]
                          },
                          {
                             "conditions": [
```

```
                                        {
                                          "fn": "booleanEquals",
                                          "argv": [
                                            {
                                              "ref": "UseDualStack"
                                            },
                                            true
                                          ]
                                        }
                                      ],
                                      "type": "tree",
                                       "rules": [
                                         {
                                           "conditions": [
                                             {
                                               "fn": "booleanEquals",
                                               "argv": [
                                                 true,
                                                 {
                                                   "fn": "getAttr",
                                                   "argv": [
                                                     {
                                                       "ref": "PartitionResult"
                                                     },

    "supportsDualStack"

                                                   ]
                                                 }
                                               ]
                                             }
                                           ],
                                           "type": "tree",
                                           "rules": [
                                             {
                                               "conditions": [],
                                               "endpoint": {
                                                 "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                 "properties": {},
                                                 "headers": {}

                                               },
                                               "type": "endpoint"
                                             }
                                           ]
                                         },
                                         {
                                           "conditions": [],
```

```
                              "error": "DualStack is enabled but this partition does not support DualStack",
                              "type": "error"
                            }
                          ]
                        },
                        {
                          "conditions": [],
                          "endpoint": {
                            "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                          },
                          "type": "endpoint"
                        }
                      ]
                    }
                  ]
                },
                {
                  "conditions": [],
                  "error": "Invalid Configuration: Missing Region",
                  "type": "error"
                }
              ]
            },
            "smithy.rules#endpointTests": {
              "testCases": [
                {
                  "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint":
{
                        "url": "https://license-manager.af-south-1.amazonaws.com"
                      }
                  },
                  "params": {
                    "Region": "af-south-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager.ap-east-1.amazonaws.com"
                    }
                  },
```

```
      "params": {
        "Region": "ap-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "ap-northeast-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-2.amazonaws.com"
        }
      },
      "params": {
        "Region": "ap-northeast-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-3.amazonaws.com"
        }
      },
      "params": {
        "Region": "ap-northeast-3",
        "UseFIPS": false,
        "UseDualStack": false
      }

    },
    {
      "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
```

```
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-1.amazonaws.com"

}
          },
          "params": {
            "Region": "ap-southeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

"documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-3.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-3",
```

```
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ca-central-1.amazonaws.com"
        }
      },

   "params": {
        "Region": "ca-central-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.eu-central-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "eu-central-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.eu-north-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "eu-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
```

```
          "endpoint": {
            "url": "https://license-manager.eu-south-1.amazonaws.com"
          }
        },
        "params": {
          "Region":
"eu-south-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.eu-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "eu-west-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
        "expect": {

          "endpoint": {
            "url": "https://license-manager.eu-west-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "eu-west-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.eu-west-3.amazonaws.com"
          }
        },
        "params": {
          "Region": "eu-west-3",
          "UseFIPS": false,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.me-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "me-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.sa-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "sa-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
```

```
          "url": "https://license-manager-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-east-2.amazonaws.com"
        }

    },
    "params": {
        "Region": "us-east-2",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-east-2.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-2",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region
us-west-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-west-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-west-1",
        "UseFIPS": false,
        "UseDualStack": false
```

```
            }
          },
          {
            "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-west-1.amazonaws.com"
              }
            },
            "params": {

        "Region": "us-west-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-west-2.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-west-2",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

    "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-west-2.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-west-2",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-east-1.api.aws"
```

```
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,

        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-northwest-1",
        "UseFIPS": false,
        "UseDualStack": false
```

```
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },

    {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.us-gov-east-1.amazonaws.com"
                }
```

```
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-west-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-west-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
```

```
                    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-fips.us-gov-east-1.api.aws"
                      }
                    },

        "params": {
                      "Region": "us-gov-east-1",
                      "UseFIPS": true,
                      "UseDualStack": true
                    }
                  },
                  {
                    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true
                    }
                  },
                  {
                    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
        disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
                      }
                    },
                    "params": {
                      "Region": "us-iso-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                    },
                    "params": {
                      "Region": "us-iso-east-1",
                      "UseFIPS": true,
```

```
                  "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
                    }
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled
but this partition does not support DualStack"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },

            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {
```

```
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
                }
              },

      "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "error": "DualStack is enabled but this partition does not support DualStack"
              },
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

      "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
```

```
                    "expect": {
                      "endpoint": {
                         "url": "https://example.com"
                      }
                    },
                    "params": {
                      "UseFIPS": false,

     "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                    "expect": {
                      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                      "error":
 "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "Missing region",
                    "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
              ],
              "version": "1.0"
            }
          }
    },
```

```
      "com.amazonaws.licensemanager#AcceptGrant": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

},
        "output": {
          "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"

}
        ],
        "traits": {
          "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
```

```
      "traits": {
        "smithy.api#input": { }
      }
    },
    "com.amazonaws.licensemanager#AcceptGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant
ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": { }
      }
    },
    "com.amazonaws.licensemanager#AccessDeniedException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {

 "code": "ServiceAccessDenied",
          "httpResponseCode": 401
        },
        "smithy.api#documentation": "<p>Access to resource denied.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 401
      }
```

```
        },
        "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
          "type": "enum",
          "members": {
            "DISTRIBUTED_GRANTS_ONLY": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
              }
            },
            "ALL_GRANTS_PERMITTED_BY_ISSUER": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#AllowedOperation": {

"type": "enum",
          "members": {
            "CREATE_GRANT": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "CreateGrant"
              }
            },
            "CHECKOUT_LICENSE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "CheckoutLicense"
              }
            },
            "CHECKOUT_BORROW_LICENSE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "CheckoutBorrowLicense"
              }
            },
            "CHECK_IN_LICENSE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "CheckInLicense"
              }
            },
            "EXTEND_CONSUMPTION_LICENSE":
```

```
{
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "ExtendConsumptionLicense"
                }
            },
            "LIST_PURCHASED_LICENSES": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "ListPurchasedLicenses"
                }
            },
            "CREATE_TOKEN": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CreateToken"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#AllowedOperationList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#AllowedOperation"
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max":
7
            }
        }
    },
    "com.amazonaws.licensemanager#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 2048
            },
            "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
        }
    },
    "com.amazonaws.licensemanager#ArnList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#Arn"
```

```
      }
    },
    "com.amazonaws.licensemanager#AuthorizationException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError":
{

          "code": "AuthorizationFailure",
          "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n       policy associated with this account.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 403
      }
    },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
      "type": "structure",
      "members": {
        "LastRunTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes automated discovery.</p>"
      }
    },

    "com.amazonaws.licensemanager#Boolean": {
      "type": "boolean",
      "traits": {
        "smithy.api#default": false
      }
    },
    "com.amazonaws.licensemanager#BorrowConfiguration": {
      "type": "structure",
      "members": {
        "AllowEarlyCheckIn": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
```

```
              "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxTimeToLiveInMinutes": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
              "smithy.api#required": {}
            }
          }

        },
          "traits": {
            "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#BoxBoolean": {
          "type": "boolean"
        },
        "com.amazonaws.licensemanager#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanager#BoxLong": {
          "type": "long"
        },
        "com.amazonaws.licensemanager#CheckInLicense": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"

          },
            {
              "target": "com.amazonaws.licensemanager#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
```

```
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
        }
      },
      "com.amazonaws.licensemanager#CheckInLicenseRequest":
 {
        "type": "structure",
        "members": {
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CheckInLicenseResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicense":
```

```
{
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
            "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
    }
},
```

```
"com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
    "type": "structure",
    "members":
{
        "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
                "smithy.api#required": {}
            }
        },
        "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
                "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
                "smithy.api#required": {}
            }
        },
        "DigitalSignatureMethod": {
            "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
            "traits": {
                "smithy.api#documentation": "<p>Digital signature method. The possible
value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
                "smithy.api#required": {}
            }
        },
        "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Node ID.</p>"
            }
        },
        "CheckoutMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
                "smithy.api#documentation": "<p>Information about constraints.</p>"
            }
        },
        "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
                "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                "smithy.api#required": {}
            }
        }
```

```
        },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },
        "EntitlementsAllowed":
{
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "SignedToken": {
          "target": "com.amazonaws.licensemanager#SignedToken",
          "traits": {
            "smithy.api#documentation": "<p>Signed token.</p>"
          }
        },
        "IssuedAt": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
          }
        },
        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target":
```

```
"com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n          <p>If the
account that created the license is the same that is performing the check out, you must\n          specify the account
as the beneficiary.</p>\n        </note>"
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
      "type": "structure",
      "members": {
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product
SKU.</p>",
            "smithy.api#required": {}
          }
        },
        "CheckoutType": {
          "target": "com.amazonaws.licensemanager#CheckoutType",
          "traits": {
            "smithy.api#documentation": "<p>Checkout type.</p>",
            "smithy.api#required": {}
          }
        },
        "KeyFingerprint": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
          }
        },
```

```
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          }
        },
        "traits": {

      "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
        "type": "structure",
        "members": {
          "CheckoutType": {
            "target": "com.amazonaws.licensemanager#CheckoutType",
            "traits": {
              "smithy.api#documentation": "<p>Checkout type.</p>"
            }
          },
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>"
            }
          },
          "EntitlementsAllowed": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
            }
          },
```

```
          "SignedToken": {
            "target": "com.amazonaws.licensemanager#SignedToken",
            "traits": {
              "smithy.api#documentation": "<p>Signed token.</p>"
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          },
          "IssuedAt": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
            }
          },
          "Expiration": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutType": {
        "type": "enum",
        "members": {
          "PROVISIONAL": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PROVISIONAL"
            }
          },
          "PERPETUAL": {
            "target": "smithy.api#Unit",
            "traits": {
```

```
              "smithy.api#enumValue": "PERPETUAL"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#ClientRequestToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 36
          }
        }
      },
      "com.amazonaws.licensemanager#ClientToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 2048
          },
          "smithy.api#pattern": "^\\S+$"
        }
      },
      "com.amazonaws.licensemanager#ConflictException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "ConflictException",
            "httpResponseCode":
409
          },
          "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 409
        }
      },
      "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
        "type": "structure",
        "members": {
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      },

  "traits": {
        "smithy.api#documentation": "<p>Details about license consumption.</p>"
      }
    },
    "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
      }
    },
    "com.amazonaws.licensemanager#ConsumptionConfiguration": {
      "type": "structure",
      "members": {
        "RenewType": {
          "target": "com.amazonaws.licensemanager#RenewType",
          "traits": {
            "smithy.api#documentation": "<p>Renewal frequency.</p>"
          }
        },
        "ProvisionalConfiguration": {
          "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
          }
        },
        "BorrowConfiguration": {
          "target": "com.amazonaws.licensemanager#BorrowConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
      }
    },
```

```
"com.amazonaws.licensemanager#CreateGrant": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#CreateGrantRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#CreateGrantResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target":
"com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n          entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
  }
},
"com.amazonaws.licensemanager#CreateGrantRequest": {
  "type": "structure",
  "members": {
    "ClientToken": {
      "target": "com.amazonaws.licensemanager#ClientToken",
      "traits": {
        "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
```

```
          "smithy.api#required": {}
        }
      },
      "GrantName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant name.</p>",
          "smithy.api#required": {}

  }
      },
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
          "smithy.api#required": {}
        }
      },
      "Principals": {
        "target": "com.amazonaws.licensemanager#PrincipalArnList",
        "traits": {
          "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n        (ARN):</p>\n    <ul>\n        <li>\n            <p>An Amazon Web Services
account, which includes only the account specified.</p>\n        </li>\n    </ul>\n    <ul>\n        <li>\n
   <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n        </li>\n    </ul>\n
<ul>\n
        <li>\n            <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n      </ul>",
          "smithy.api#required": {}
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
        "traits": {
          "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
```

```
          }
        },

    "com.amazonaws.licensemanager#CreateGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
    "com.amazonaws.licensemanager#CreateGrantVersion":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
```

```
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
      "type": "structure",
      "members": {
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
          }
        },
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant name.</p>"
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
```

```
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "StatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Grant status reason.</p>"
          }
        },
        "SourceVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the grant.</p>"
          }
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
            "smithy.api#documentation": "<p>The options specified for the grant.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
```

```
                    "traits": {
                        "smithy.api#documentation": "<p>New version of the grant.</p>"
                    }
                }
            },
            "traits":
{
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicense": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RedirectException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Creates a license.</p>"
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
            "type": "operation",
            "input": {
```

```
            "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of
licenses purchased and used.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Name of the license configuration.</p>",
                    "smithy.api#required": {}
                }
            },
            "Description": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Description of the license configuration.</p>"
```

```
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n          blocks the launch of new instances.</p>"
          }
        },
        "LicenseRules": {
          "target":
 "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n         vary by dimension, as follows.</p>\n
<ul>\n         <li>\n            <p>\n              <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
    <code>licenseAffinityToHost</code> |\n              <code>maximumCores</code> |
<code>minimumCores</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n            <code>maximumCores</code> |
<code>minimumCores</code> |\n            <code>maximumSockets</code> | <code>minimumSockets</code> |\n
    <code>maximumVcpus</code> | <code>minimumVcpus</code>\n            </p>\n         </li>\n
<li>\n
 <p>\n              <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n              <code>maximumSockets</code> |
<code>minimumSockets</code>\n            </p>\n         </li>\n         <li>\n            <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n              <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n            </p>\n         </li>\n         </ul>\n         <p>The unit for
<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n         values for
<code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n
<code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n
<code>True</code> and <code>False</code>.</p>"
          }
        },
```

```
    "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
                "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
            }
        },
        "DisassociateWhenNotFound": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
                "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
            }
        },
        "ProductInformationList": {
            "target": "com.amazonaws.licensemanager#ProductInformationList",
            "traits": {
                "smithy.api#documentation": "<p>Product information.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
    "type": "structure",
    "members": {
        "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
    },
    "errors": [
```

```
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
               },
              {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
              {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              },
              {
                "target": "com.amazonaws.licensemanager#ValidationException"
              }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {

              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
                "smithy.api#required": {}
              }
            },
            "SourceLicenseContext": {
              "target": "com.amazonaws.licensemanager#LicenseConversionContext",
              "traits": {
                "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n        For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
                "smithy.api#required": {}
              }
            },
            "DestinationLicenseContext":
 {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
    "type": "structure",
    "members": {
        "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

    "traits": {
            "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
        }
    }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "ReportGeneratorName":
{
          "target": "com.amazonaws.licensemanager#ReportGeneratorName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>",
            "smithy.api#required": {}
          }
        },
        "Type": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n          <ul>\n          <li>\n             <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n          </li>\n          <li>\n          <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n          </li>\n
</ul>",
            "smithy.api#required": {}
          }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
            "smithy.api#required": {}
          }
```

```
            },
            "ReportFrequency": {
              "target": "com.amazonaws.licensemanager#ReportFrequency",
              "traits": {
                "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
                "smithy.api#required": {}
              }
            },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientRequestToken",
              "traits": {
                "smithy.api#documentation": "<p>Unique,
 case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanager#TagList",
              "traits": {
                "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
          "type": "structure",
          "members": {
            "LicenseManagerReportGeneratorArn":
 {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
```

```
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product
name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>",
            "smithy.api#required": {}
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#Issuer",
          "traits": {
            "smithy.api#documentation": "<p>License issuer.</p>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region for the license.</p>",
            "smithy.api#required": {}

  }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
            "smithy.api#required": {}
```

```
                }
              },
              "Entitlements": {
                "target": "com.amazonaws.licensemanager#EntitlementList",
                "traits": {
                  "smithy.api#documentation": "<p>License entitlements.</p>",
                  "smithy.api#required": {}
                }
              },
              "Beneficiary": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>License beneficiary.</p>",
                  "smithy.api#required": {}
                }
              },
              "ConsumptionConfiguration": {
                "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                "traits": {
                  "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n          running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n          usage.</p>",
                  "smithy.api#required": {}
                }
              },
              "LicenseMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                  "smithy.api#documentation": "<p>Information about the license.</p>"
                }
              },
              "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                  "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                  "smithy.api#required": {}
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#CreateLicenseResponse": {
            "type": "structure",
            "members": {
              "LicenseArn": {
```

```
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersion": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target":
"com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product name.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits":
{
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

                    "smithy.api#documentation": "<p>Home Region of the license.</p>",
                    "smithy.api#required": {}
                }
            },
            "Validity": {
                "target": "com.amazonaws.licensemanager#DatetimeRange",
                "traits": {
                    "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-UTC format.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license.</p>"
                }
            },
            "Entitlements": {
                "target": "com.amazonaws.licensemanager#EntitlementList",
                "traits": {
                    "smithy.api#documentation": "<p>License entitlements.</p>",
                    "smithy.api#required": {}
                }
            },
            "ConsumptionConfiguration": {
                "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional configuration for workloads\n   running with continuous connectivity. Choose a borrow configuration for workloads with offline\n   usage.</p>",
                    "smithy.api#required": {}
                }
            },

    "Status": {
                "target": "com.amazonaws.licensemanager#LicenseStatus",
                "traits": {
                    "smithy.api#documentation": "<p>License status.</p>",
                    "smithy.api#required": {}
                }
            },
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                    "smithy.api#required": {}

```
              }
            },
            "SourceVersion": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Current version of the license.</p>"
              }
            }
          },
          "traits": {

"smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>License ARN.</p>"
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>New version of the license.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanager#LicenseStatus",
              "traits": {
                "smithy.api#documentation": "<p>License status.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output":
{}
          }
        },
        "com.amazonaws.licensemanager#CreateToken": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CreateTokenRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CreateTokenResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },

      {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateTokenRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits":
 {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
                "smithy.api#required": {}
            }
        },
        "RoleArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n      License Manager does not check whether the roles are in use.</p>"
            }
          },
          "ExpirationInDays": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
              "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
            }
          },
          "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
            "traits": {
              "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n      to the amr claim of the JWT token.</p>"
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateTokenResponse": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",

          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
        "TokenType": {
          "target": "com.amazonaws.licensemanager#TokenType",
          "traits": {
            "smithy.api#documentation": "<p>Token type.</p>"
          }
        },
        "Token": {
```

```
            "target": "com.amazonaws.licensemanager#TokenString",
            "traits": {
                "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licensemanager#DateTime": {
    "type": "timestamp"
},
"com.amazonaws.licensemanager#DatetimeRange": {
    "type": "structure",
    "members": {
        "Begin": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>Start of the time range.</p>",
                "smithy.api#required": {}
            }
        },
        "End": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
                "smithy.api#documentation": "<p>End of the time range.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
    }
},
"com.amazonaws.licensemanager#DeleteGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
            }
        },
        "com.amazonaws.licensemanager#DeleteGrantRequest": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                        "smithy.api#required": {}
                    }
                },
                "StatusReason": {
                    "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                    "traits": {
                        "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Current version of the
grant.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
```

```
                "smithy.api#input": {}
              }
            },
          "com.amazonaws.licensemanager#DeleteGrantResponse": {
            "type": "structure",
            "members": {
              "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                  "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
              },
              "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                  "smithy.api#documentation": "<p>Grant status.</p>"
                }
              },
              "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation":
"<p>Grant version.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#DeleteLicense": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
              {
                "target": "com.amazonaws.licensemanager#ConflictException"
              },
```

```
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"

        },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
        },

    "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
```

```
              "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n          <p>You cannot
delete a license configuration that is in use.</p>"
            }
          },
      "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
          "type": "structure",
          "members":
{
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>ID of the license configuration.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
      "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {

  "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
```

```
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                "smithy.api#required": {}
              }
            },
            "SourceVersion": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Current version of the license.</p>",
                "smithy.api#required":
{}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseResponse": {
          "type": "structure",
          "members": {
            "Status": {
              "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
              "traits": {
                "smithy.api#documentation": "<p>License status.</p>"
              }
            },
            "DeletionDate": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#DeleteToken": {
          "type": "operation",
           "input": {
            "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
          },
          "errors": [
            {
```

```
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },

    {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenRequest": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanager#DigitalSignatureMethod": {
        "type": "enum",
        "members": {
          "JWT_PS384": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "JWT_PS384"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#Entitlement": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement name.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
            }
          },
          "MaxCount": {
            "target": "com.amazonaws.licensemanager#Long",
            "traits": {
              "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
            }
          },
          "Overage": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement unit.</p>",
              "smithy.api#required": {}
            }
          }
```

```
        },
        "AllowCheckIn": {
          "target":
"com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementData": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement data name.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement
data value.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement data unit.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementDataList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementData"
      }
    },
```

```
"com.amazonaws.licensemanager#EntitlementDataUnit": {
    "type": "enum",
    "members": {
        "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {

"smithy.api#enumValue": "Count"
            }
        },
        "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "None"
            }
        },
        "SECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Seconds"
            }
        },
        "MICROSECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Microseconds"
            }
        },
        "MILLISECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Milliseconds"
            }
        },
        "BYTES": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Bytes"
            }
        },
        "KILOBYTES": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Kilobytes"
            }
        },
        "MEGABYTES": {
            "target": "smithy.api#Unit",
```

```
        "traits": {
          "smithy.api#enumValue": "Megabytes"
        }
      },
      "GIGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes"
        }
      },
      "TERABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabytes"
        }
      },
      "BITS":
{
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits"
        }
      },
      "KILOBITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits"
        }
      },
      "MEGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabits"
        }
      },
      "GIGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits"
        }
      },
      "TERABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits"

}
      },
```

```
"PERCENT": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Percent"
  }
},
"BYTES_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Bytes/Second"
  }
},
"KILOBYTES_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Kilobytes/Second"
  }
},
"MEGABYTES_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Megabytes/Second"
  }
},
"GIGABYTES_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Gigabytes/Second"
  }
},
"TERABYTES_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Terabytes/Second"
  }
},
"BITS_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Bits/Second"
  }
},
"KILOBITS_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Kilobits/Second"
  }
},
```

```
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
          }
        },
        "TERABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits/Second"
          }
        },
        "COUNT_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count/Second"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Entitlement"
      }
    },
    "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#EntitlementUnit": {
```

```
"type": "enum",
"members": {
  "COUNT": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Count"
    }
  },
  "NONE": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "None"
    }
  },
  "SECONDS":
{
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Seconds"
      }
  },
  "MICROSECONDS": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Microseconds"
    }
  },
  "MILLISECONDS": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Milliseconds"
    }
  },
  "BYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Bytes"
    }
  },
  "KILOBYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Kilobytes"
    }
  },
  "MEGABYTES": {
    "target": "smithy.api#Unit",
    "traits": {
```

```
      "smithy.api#enumValue": "Megabytes"
    }
  },
  "GIGABYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Gigabytes"
    }
  },
  "TERABYTES": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Terabytes"
    }
  },
  "BITS": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Bits"
    }
  },
  "KILOBITS": {
    "target": "smithy.api#Unit",
    "traits": {

      "smithy.api#enumValue": "Kilobits"
    }
  },
  "MEGABITS": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Megabits"
    }
  },
  "GIGABITS": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Gigabits"
    }
  },
  "TERABITS": {
    "target": "smithy.api#Unit",
    "traits": {
      "smithy.api#enumValue": "Terabits"
    }
  },
  "PERCENT": {
    "target": "smithy.api#Unit",
```

```
          "traits": {
            "smithy.api#enumValue": "Percent"
          }
        },
        "BYTES_PER_SECOND": {
          "target":
"smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bytes/Second"
          }
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {

"smithy.api#enumValue": "Terabytes/Second"
          }
        },
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
```

```
      "MEGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabits/Second"
        }
      },
      "GIGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits/Second"
        }
      },

  "TERABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits/Second"
        }
      },
      "COUNT_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Count/Second"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#EntitlementUsage": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement usage name.</p>",
          "smithy.api#required": {}
        }
      },
      "ConsumedValue": {
        "target": "com.amazonaws.licensemanager#String",

      "traits": {
          "smithy.api#documentation": "<p>Resource usage consumed.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxCount": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
```

```
            "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList":
{
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#EntitlementUsage"
        }
      },
      "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
          }
        },
        "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
          "type": "structure",
          "members": {
            "LicenseConsumptionToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License consumption token.</p>",
                "smithy.api#required": {}
              }

          }
            },
            "DryRun": {
              "target": "com.amazonaws.licensemanager#Boolean",
              "traits": {
                "smithy.api#default": false,
                "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
          "type": "structure",
          "members": {
            "LicenseConsumptionToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License consumption token.</p>"
              }
            },

            "Expiration": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
              }
```

```
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#FailedDependencyException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        },
        "ErrorCode": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "FailedDependency",
          "httpResponseCode": 424
        },
        "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 424
      }
    },
    "com.amazonaws.licensemanager#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#FilterName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
          }
        },
        "Values": {
          "target": "com.amazonaws.licensemanager#FilterValues",
          "traits": {
            "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter
name and value pair that is used to return more specific results from a\n          describe operation. Filters can be used
to match a set of resources by specific criteria,\n          such as tags, attributes, or IDs.</p>"
```

```
      }
    },
    "com.amazonaws.licensemanager#FilterLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "FilterLimitExceeded",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Filter"
      }
    },
    "com.amazonaws.licensemanager#FilterName": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#FilterValue": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#FilterValues": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#FilterValue",
        "traits": {
          "smithy.api#xmlName": "item"
        }
      }
    },
    "com.amazonaws.licensemanager#Filters": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Filter",
        "traits": {
          "smithy.api#xmlName": "item"
        }
      }
```

```
        },
      "com.amazonaws.licensemanager#GetAccessToken": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
               "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
               "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
               "target": "com.amazonaws.licensemanager#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
 tokens\n       are valid for one hour.</p>"
         }
      },
      "com.amazonaws.licensemanager#GetAccessTokenRequest": {
         "type": "structure",
         "members": {
            "Token": {
               "target": "com.amazonaws.licensemanager#TokenString",
               "traits": {
                  "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
                  "smithy.api#required": {}
               }
            },
            "TokenProperties": {
               "target": "com.amazonaws.licensemanager#MaxSize3StringList",
               "traits": {
                  "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
               }
```

```
            }
          },
          "traits": {
            "smithy.api#input": { }
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
          "type": "structure",
          "members": {
            "AccessToken": {
              "target": "com.amazonaws.licensemanager#TokenString",
              "traits": {
                "smithy.api#documentation": "<p>Temporary access token.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": { }
          }
        },
        "com.amazonaws.licensemanager#GetGrant": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#GetGrantRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#GetGrantResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"

            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```json
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetGrantRequest": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrantResponse": {
      "type": "structure",
      "members": {
        "Grant": {
          "target": "com.amazonaws.licensemanager#Grant",
          "traits": {
            "smithy.api#documentation": "<p>Grant details.</p>"
          }
        }
      },
      "traits": {

"smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicense": {
      "type": "operation",
      "input": {
```

            "target": "com.amazonaws.licensemanager#GetLicenseRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetLicenseConfiguration": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits":
{
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurationId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
          }
        },
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
          "smithy.api#documentation": "<p>Name of the license configuration.</p>"
        }
      },
      "Description": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Description of the license configuration.</p>"
        }
      },
      "LicenseCountingType": {
        "target": "com.amazonaws.licensemanager#LicenseCountingType",
        "traits": {
          "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",

        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License configuration status.</p>"
        }
      }

    },
      "OwnerAccountId": {
```

```
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
          }
        },
        "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
          }
        },
        "ManagedResourceSummaryList": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
           "traits": {
            "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>Product information.</p>"
          }
        },
        "AutomatedDiscoveryInformation": {
          "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
          "traits": {
            "smithy.api#documentation": "<p>Automated discovery information.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#GetLicenseConversionTask": {
```

```json
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {

    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
        "type": "structure",
        "members": {
```

```
"LicenseConversionTaskId": {
    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
    "traits": {
        "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
    }
},
"ResourceArn": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
    }
},
"SourceLicenseContext": {
    "target": "com.amazonaws.licensemanager#LicenseConversionContext",
    "traits": {
        "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
    }
},
"DestinationLicenseContext": {
    "target": "com.amazonaws.licensemanager#LicenseConversionContext",
    "traits": {
        "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
    }
},
"StatusMessage": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
    }
},
"Status": {
    "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
    "traits": {
        "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
    }
},

"StartTime": {
    "target": "com.amazonaws.licensemanager#DateTime",
    "traits": {
        "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
    }
},
"LicenseConversionTime": {
    "target": "com.amazonaws.licensemanager#DateTime",
    "traits": {
```

```
                    "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

            },
            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
              "smithy.api#required":
{}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
      "type": "structure",
      "members": {
        "ReportGenerator": {
          "target": "com.amazonaws.licensemanager#ReportGenerator",
          "traits": {
            "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
```

```
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseResponse": {
        "type": "structure",
        "members": {
          "License": {
            "target": "com.amazonaws.licensemanager#License",
            "traits": {
              "smithy.api#documentation": "<p>License details.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },

"com.amazonaws.licensemanager#GetLicenseUsage": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
```

```
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {

    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
            "type": "structure",
            "members": {
                "LicenseUsage": {
                    "target":
"com.amazonaws.licensemanager#LicenseUsage",
                    "traits": {
                        "smithy.api#documentation": "<p>License usage details.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#GetServiceSettings": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
            },
            "output": {
```

```
              "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
            },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
             },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "S3BucketArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"

  }
            },
            "SnsTopicArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
              }
            },
            "OrganizationConfiguration": {
              "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
              "traits": {
```

```
                    "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n        cross-account discovery.</p>"
                }
            },
            "EnableCrossAccountsDiscovery": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
                }

        },
            "LicenseManagerResourceShareArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n        provides member accounts with access to this share.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#Grant": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantName": {

 "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant name.</p>",
                    "smithy.api#required": {}
                }
            },
            "ParentArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Parent ARN.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>License ARN.</p>",
            "smithy.api#required": {}
          }
        },
        "GranteePrincipalArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {

"smithy.api#documentation": "<p>The grantee principal ARN.</p>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantStatus": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Grant status reason.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Granted operations.</p>",
            "smithy.api#required": {}
          }
```

```
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
            "smithy.api#documentation": "<p>The options specified for the grant.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Describes a grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#GrantList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Grant"
      }
    },
    "com.amazonaws.licensemanager#GrantStatus": {
      "type": "enum",
      "members": {
        "PENDING_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_WORKFLOW"
          }
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_ACCEPT"
          }
        },
        "REJECTED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"

          }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
```

```
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "FAILED_WORKFLOW"
              }
            },
            "DELETED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "DELETED"
              }
            },
            "PENDING_DELETE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "PENDING_DELETE"
              }
            },
            "DISABLED": {
              "target": "smithy.api#Unit",
              "traits":
{
                "smithy.api#enumValue": "DISABLED"
              }
            },
            "WORKFLOW_COMPLETED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "WORKFLOW_COMPLETED"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#GrantedLicense": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
              }
            },
            "LicenseName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License name.</p>"
              }
```

```
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#IssuerDetails",
          "traits": {
            "smithy.api#documentation": "<p>Granted license issuer.</p>"
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>Granted license status.</p>"
          }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
ISO8601-UTC format.</p>"
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
```

```
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>"
        }
      },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
        }
      },
      "LicenseMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
          "smithy.api#documentation": "<p>Granted license metadata.</p>"
        }
      },
      "CreateTime": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

}
      },
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Version of the granted license.</p>"
        }
      },
      "ReceivedMetadata": {
        "target": "com.amazonaws.licensemanager#ReceivedMetadata",
        "traits": {
          "smithy.api#documentation": "<p>Granted license received metadata.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
    }
  },
  "com.amazonaws.licensemanager#GrantedLicenseList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#GrantedLicense"
    }
  },
  "com.amazonaws.licensemanager#ISO8601DateTime":
```

```
  {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-
1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
        }
      },
      "com.amazonaws.licensemanager#Integer": {
        "type": "integer"
      },
      "com.amazonaws.licensemanager#InvalidParameterValueException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidParameterValueProvided",
            "httpResponseCode": 400
          },
          "smithy.api#documentation":
"<p>One or more parameter values are not valid.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#InvalidResourceStateException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidResourceState",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n    <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
          "smithy.api#error": "client",
```

```
          "smithy.api#httpError": 400
        }
    },

  "com.amazonaws.licensemanager#InventoryFilter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Condition": {
          "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
          "traits": {
            "smithy.api#documentation": "<p>Condition of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }

    },
      "traits": {
        "smithy.api#documentation": "<p>An inventory filter.</p>"
      }
    },
  "com.amazonaws.licensemanager#InventoryFilterCondition": {
      "type": "enum",
      "members": {
        "EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EQUALS"
          }
        },
        "NOT_EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "NOT_EQUALS"
          }
        },
```

```json
        "BEGINS_WITH": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "BEGINS_WITH"
          }
        },
        "CONTAINS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CONTAINS"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#InventoryFilter"
      }
    },
    "com.amazonaws.licensemanager#Issuer": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Issuer name.</p>",
            "smithy.api#required": {}
          }
        },
        "SignKey": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
 of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#IssuerDetails": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>"
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#License": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

      "smithy.api#documentation": "<p>License name.</p>"
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product name.</p>"
            }
          },
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
          "smithy.api#documentation": "<p>Product SKU.</p>"
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#IssuerDetails",
        "traits": {
          "smithy.api#documentation": "<p>License issuer.</p>"
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {

"smithy.api#documentation": "<p>Home Region of the license.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
        "traits": {
          "smithy.api#documentation": "<p>License status.</p>"
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License beneficiary.</p>"
        }
      },
      "Entitlements": {

  "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>"
        }
      },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
        }
      },
```

```
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>License metadata.</p>"
            }
          },
          "CreateTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>License
creation time.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfiguration": {
      "type": "structure",
      "members": {
        "LicenseConfigurationId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
          }
        },
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
           "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules.</p>"
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the license configuration.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "com.amazonaws.licensemanager#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

        }
        },
        "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
          }
        },
        "ManagedResourceSummaryList": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>Product information.</p>"
          }
        },
        "AutomatedDiscoveryInformation": {

          "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
          "traits": {
            "smithy.api#documentation": "<p>Automated discovery information.</p>"
          }
        }
      }
      },
      "traits": {
        "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n        consumed and enforced by License Manager. Components include specifications for
the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n        must be associated with a
host), and the number of licenses purchased and used.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
      "type": "structure",
      "members": {
        "ResourceArn":
{
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
```

```
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Type of server resource.</p>"
        }
      },
      "ResourceOwnerId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
        }
      },
      "AssociationTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
        }
      },
      "AmiAssociationScope": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
    }
  },
  "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
    "type": "enum",
    "members":
{
      "AVAILABLE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "AVAILABLE"
        }
```

```
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISABLED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits":
{
            "smithy.api#documentation": "<p>Type of resource.</p>"
          }
        },
        "ResourceStatus": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the resource.</p>"
          }
        },
        "ResourceOwnerId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
          }
        },
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
          }
        },

    "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
        }
    },
    "com.amazonaws.licensemanager#LicenseConfigurations": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#LicenseConfiguration"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
        "members": {
            "UsageOperation": {
                "target": "com.amazonaws.licensemanager#UsageOperation",
                "traits": {
                    "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n        </p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {

    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
            }
```

```
      },
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
        }
      },
      "SourceLicenseContext": {
        "target": "com.amazonaws.licensemanager#LicenseConversionContext",
        "traits": {
          "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
        }
      },
      "DestinationLicenseContext": {
        "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
        "traits": {
          "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
        "traits": {
          "smithy.api#documentation": "<p>The status of the conversion task.</p>"
        }
      },
      "StatusMessage": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
        }
      },
      "StartTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {
          "smithy.api#documentation":
 "<p>The time the conversion task was started at.</p>"
        }
      },
      "LicenseConversionTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {
          "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
        }
```

        },
        "EndTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskId": {
      "type": "string",
      "traits": {

      "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
      "type": "enum",
      "members": {
        "IN_PROGRESS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "IN_PROGRESS"
          }
        },
        "SUCCEEDED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUCCEEDED"
          }
        },
        "FAILED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTasks":

```
{
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTask"
        }
    },
    "com.amazonaws.licensemanager#LicenseCountingType": {
        "type": "enum",
        "members": {
            "VCPU": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "vCPU"
                }
            },
            "INSTANCE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Instance"
                }
            },
            "CORE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Core"
                }
            },
            "SOCKET": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Socket"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#LicenseDeletionStatus": {
        "type": "enum",
        "members": {
            "PENDING_DELETE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "PENDING_DELETE"
                }
            },
            "DELETED": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DELETED"
```

```
                }
            }
        }
    },
    "com.amazonaws.licensemanager#LicenseList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#License"
        }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailure": {

        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
                }
            },
            "ResourceType": {
                "target": "com.amazonaws.licensemanager#ResourceType",
                "traits": {
                    "smithy.api#documentation": "<p>Resource type.</p>"
                }
            },
            "ErrorMessage": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Error message.</p>"
                }
            },
            "FailureTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {

"smithy.api#documentation": "<p>Failure time.</p>"
                }
            },
            "OperationName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Name of the operation.</p>"
                }
            },
            "ResourceOwnerId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
```

```
                   "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
                }
            },
            "OperationRequestedBy": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
                }
            },
            "MetadataList": {

                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Reserved.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailureList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
        }
    },
    "com.amazonaws.licensemanager#LicenseSpecification": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

                    "smithy.api#required": {}
                }
            },
            "AmiAssociationScope": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
                }
            }
        },
```

```
    "traits": {
      "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
    }
},
"com.amazonaws.licensemanager#LicenseSpecifications": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanager#LicenseSpecification"
  }
},
"com.amazonaws.licensemanager#LicenseStatus": {
  "type": "enum",
  "members": {
    "AVAILABLE": {
      "target": "smithy.api#Unit",

      "traits": {
          "smithy.api#enumValue": "AVAILABLE"
        }
      },
    "PENDING_AVAILABLE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "PENDING_AVAILABLE"
      }
    },
    "DEACTIVATED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "DEACTIVATED"
      }
    },
    "SUSPENDED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "SUSPENDED"
      }
    },
    "EXPIRED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "EXPIRED"
      }
    },

  "PENDING_DELETE": {
      "target": "smithy.api#Unit",
      "traits": {
```

```
              "smithy.api#enumValue": "PENDING_DELETE"
            }
          },
          "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DELETED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseUsage": {
        "type": "structure",
        "members": {
          "EntitlementUsages": {
            "target": "com.amazonaws.licensemanager#EntitlementUsageList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlement usages.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

        }
      },
      "com.amazonaws.licensemanager#LicenseUsageException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "LicenseUsageFailure",
            "httpResponseCode": 412
          },
          "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 412
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
```

```
        },
        "output": {

    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
            }
```

```
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurationAssociations": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
          "traits": {
            "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
          }
        },
        "NextToken": {

 "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrants": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
      "type": "structure",
      "members": {
        "GrantArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {

          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>LicenseArn</code>\n            </p>\n
          </li>\n          <li>\n            <p>\n              <code>GrantStatus</code>\n            </p>\n          </li>\n
   <li>\n            <p>\n              <code>GranteePrincipalARN</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>ProductSKU</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>LicenseIssuerName</code>\n
          </p>\n          </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
```

```
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
      "type": "structure",
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",

    "traits": {
            "smithy.api#documentation": "<p>Distributed grant details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
      },
      "errors": [
        {

    "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
    "type": "structure",
    "members": {
      "LicenseConfigurationArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
    "type": "structure",
    "members": {
      "LicenseOperationFailureList": {
        "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
      },

  "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license configurations
for your account.</p>"
      }
```

```
        },
      "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n        Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n        Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n
<p>\n        <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n        Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
    </li>\n        <li>\n        <p>\n        <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
        "type": "structure",
        "members": {
```

```
      "LicenseConfigurations": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurations",
        "traits": {
          "smithy.api#documentation": "<p>Information about the license configurations.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",

     "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
```

```
        "type": "structure",
        "members": {
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {

      "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>\n        Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n      </p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTasks": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
```

"com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target":
"com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
    }
},
"com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n   \n   \n   <p>\n   <code>LicenseConfigurationArn</code>\n

```
</p>\n        </li>\n        </ul>"
              }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits":
{

              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
        "type": "structure",
        "members": {
          "ReportGenerators": {
            "target": "com.amazonaws.licensemanager#ReportGeneratorList",
            "traits": {
              "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }

        }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
        },
```

```
"output": {
  "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanager#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanager#AuthorizationException"
  },
  {
    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
  },
  {
    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
  },
  {
    "target": "com.amazonaws.licensemanager#ServerInternalException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
}
},
"com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanager#BoxInteger",
      "traits": {

        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
      }
    },
    "NextToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    }
```

```
          },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {
        "LicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                        "smithy.api#required": {}
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
            "type": "structure",
            "members": {
                "Licenses": {
                    "target": "com.amazonaws.licensemanager#LicenseList",
                    "traits": {
                        "smithy.api#documentation": "<p>License details.</p>"
                    }
                },

"NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
```

```
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenses": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicensesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicensesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },
```

```
    "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n    <ul>\n       <li>\n         <p>\n            <code>Beneficiary</code>\n         </p>\n
      </li>\n       <li>\n         <p>\n            <code>ProductSKU</code>\n         </p>\n       </li>\n
   <li>\n       <p>\n            <code>Fingerprint</code>\n         </p>\n       </li>\n       <li>\n
<p>\n            <code>Status</code>\n         </p>\n       </li>\n    </ul>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token
for the next set of results.</p>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#ListLicensesResponse": {
    "type": "structure",
    "members": {
        "Licenses": {
            "target": "com.amazonaws.licensemanager#LicenseList",
            "traits": {
                "smithy.api#documentation": "<p>License details.</p>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
```

```
        },
        "com.amazonaws.licensemanager#ListReceivedGrants": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n        recipient as this Amazon Web Services account, your organization, or an organizational unit\n
(OU) to which this member account belongs.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
          },
          "errors": [
            {
```

```
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target":
"com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
    "type": "structure",
    "members": {
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
          "smithy.api#required": {}
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
    <ul>\n        <li>\n          <p>\n            <code>ParentArn</code>\n          </p>\n        </li>\n
<li>\n          <p>\n            <code>GranteePrincipalArn</code>\n          </p>\n        </li>\n      </ul>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
```

```
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
        "type": "structure",
        "members": {
          "Grants": {
            "target": "com.amazonaws.licensemanager#GrantList",
            "traits": {
              "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
        "type": "structure",
        "members": {
          "GrantArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
```

supported:</p>\n   \n   \n   <p>\n   <code>ProductSKU</code>\n   </p>\n
        \n   \n   <p>\n   <code>LicenseIssuerName</code>\n   </p>\n
\n   \n   <p>\n   <code>LicenseArn</code>\n   </p>\n   \n
\n   <p>\n   <code>GrantStatus</code>\n   </p>\n   \n   \n
<p>\n

                <code>GranterAccountId</code>\n   </p>\n   \n   "
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
        "type": "structure",
        "members": {
            "Grants": {
                "target": "com.amazonaws.licensemanager#GrantList",
                "traits": {
                    "smithy.api#documentation": "<p>Received grant details.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedLicenses": {
        "type": "operation",
        "input": {

```
              "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
          },
          "output": {
              "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
          },
          "errors": [
              {
                  "target": "com.amazonaws.licensemanager#AccessDeniedException"
               },
              {
                  "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
              {
                  "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                  "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                  "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
              },
              {
                  "target": "com.amazonaws.licensemanager#ServerInternalException"
              },
              {
                  "target": "com.amazonaws.licensemanager#ValidationException"
              }
          ],
          "traits": {
              "smithy.api#documentation": "<p>Lists received licenses.</p>"
          }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
  {
          "type": "operation",
          "input": {
              "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
          },
          "output": {
              "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
          },
          "errors": [
              {
                  "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
                  "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
```

```
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {

    "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanager#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>Beneficiary</code>\n            </p>\n
        </li>\n          <li>\n            <p>\n              <code>ProductSKU</code>\n
        </p>\n          </li>\n        </ul>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
```

```
            }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
            "type": "structure",
            "members": {
                "Licenses": {
                    "target": "com.amazonaws.licensemanager#GrantedLicenseList",
                    "traits": {
                        "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
            "type": "structure",
            "members": {
                "LicenseArns": {
                    "target": "com.amazonaws.licensemanager#ArnList",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
                    }
                },

                "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>ProductSKU</code>\n                </p>\n
            </li>\n            <li>\n                <p>\n                    <code>Status</code>\n                </p>\n            </li>\n
<li>\n                <p>\n                    <code>Fingerprint</code>\n                </p>\n            </li>\n            <li>\n
<p>\n                    <code>IssuerName</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Beneficiary</code>\n                </p>\n            </li>\n        </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
```

```
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#GrantedLicenseList",
          "traits": {
            "smithy.api#documentation": "<p>Received license details.</p>"
          }
        },

        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListResourceInventory": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
```

```
        {
          "target": "com.amazonaws.licensemanager#FailedDependencyException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#InventoryFilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n          are supported:</p>\n          <ul>\n             <li>\n                <p>\n                   <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
             </li>\n             <li>\n                <p>\n                   <code>application_name</code> - The name of the
application. \n            Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
   </li>\n             <li>\n                <p>\n                   <code>license_included</code> - The type of license included. \n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n            Possible values
are <code>sql-server-enterprise</code> | \n                <code>sql-server-standard</code> | \n                <code>sql-
server-web</code> | \n                <code>windows-server-datacenter</code>.</p>\n             </li>\n             <li>\n
```

`<p>\n   <code>platform</code> - The platform of the resource.\n   Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n   \n   \n   <p>\n   <code>resource_id</code> - The ID of the resource. \n   Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>tag:<key></code> - The key/value combination of a tag assigned \n   to the resource. Logical operators are <code>EQUALS</code> (single account) or\n   <code>EQUALS</code> | <code>NOT_EQUALS</code> (cross account).</p>\n   \n   "`

```
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
            "type": "structure",
            "members": {
              "ResourceInventoryList": {
                "target": "com.amazonaws.licensemanager#ResourceInventoryList",
                "traits": {
                  "smithy.api#documentation": "<p>Information
about the resources.</p>"
                }
              },
              "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#ListTagsForResource": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
```

```
                "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
              {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

          "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
              "target": "com.amazonaws.licensemanager#TagList",
              "traits": {
                "smithy.api#documentation": "<p>Information about the tags.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListTokens": {
```

            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListTokensRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#ListTokensResponse"
            },
            "errors":

[

                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists your tokens.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListTokensRequest": {
            "type": "structure",
            "members": {
                "TokenIds": {
                    "target": "com.amazonaws.licensemanager#StringList",
                    "traits": {

                        "smithy.api#documentation": "<p>Token IDs.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    \n   \n   <p>\n   <code>LicenseArns</code>\n   </p>\n   \n
"
                    }
                },
                "NextToken": {

```json
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
          "Tokens": {
            "target": "com.amazonaws.licensemanager#TokenList",
            "traits": {
              "smithy.api#documentation": "<p>Received token details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
          {
```

```
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying license\n        consumption details by resource at a selected point in time. Use this action to audit the\n        current license consumption for any license inventory and configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
```

```
                "target": "com.amazonaws.licensemanager#Filters",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n            <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n
 <li>\n          <p>\n            <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n          <p>\n
    <code>resourceAccount</code> - The ID of the account that owns the resource. \n        Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
        "type": "structure",
        "members": {

 "LicenseConfigurationUsageList": {
                "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license configurations.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#Location": {
        "type": "string"
    },
    "com.amazonaws.licensemanager#Long": {
        "type": "long"
    },
    "com.amazonaws.licensemanager#ManagedResourceSummary": {
        "type": "structure",
```

```
    "members": {
        "ResourceType":
{
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
                "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
            }
        },
        "AssociationCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
                "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
    }
},
"com.amazonaws.licensemanager#ManagedResourceSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
    }
},
"com.amazonaws.licensemanager#MaxSize100": {

"type": "integer",
    "traits": {
        "smithy.api#range": {
            "min": 1,
            "max": 100
        }
    }
},
"com.amazonaws.licensemanager#MaxSize3StringList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 3
        }
    }
},
"com.amazonaws.licensemanager#Message": {
```

```
      "type": "string"
    },
    "com.amazonaws.licensemanager#Metadata": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The key name.</p>"
          }
        },

  "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The value.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
      }
    },
    "com.amazonaws.licensemanager#MetadataList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Metadata"
      }
    },
    "com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#Options": {
      "type": "structure",
      "members": {
        "ActivationOverrideBehavior": {
          "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
```

```
          "traits": {
                "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n          Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n          <ul>\n          <li>\n          <p>As a license administrator, you can
optionally specify an\n          <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
</li>\n          <li>\n          <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n          account in your
organization.</p>\n          </li>\n          <li>\n          <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n          <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n          are activating the grant.</p>\n          </li>\n          </ul>\n          <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n          <dd>\n          <p>Use this value to activate a grant
without replacing any member accounts\n          active grants for the same product.</p>\n          </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n          <dd>\n          <p>Use this value to activate a
grant and disable other active grants in any\n          member accounts
 for the same product. This action will also replace their\n          previously activated grants with this activated
grant.</p>\n          </dd>\n          </dl>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n          override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#OrganizationConfiguration": {
      "type": "structure",
      "members": {
        "EnableIntegration": {
          "target": "com.amazonaws.licensemanager#Boolean",
          "traits": {
            "smithy.api#default": false,
            "smithy.api#documentation":
 "<p>Enables Organizations integration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
      }
    },
    "com.amazonaws.licensemanager#PrincipalArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
```

```
          },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 1
          }
        }
      },
    "com.amazonaws.licensemanager#ProductInformation": {
      "type": "structure",
      "members": {
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Resource type. The
 possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductInformationFilterList": {
          "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
          "traits": {
            "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n        <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n        <p>The following filters and are supported when
the resource type \n        is <code>SSM_MANAGED</code>:</p>\n
      <ul>\n          <li>\n             <p>\n                <code>Application Name</code> - The name of the
application.\n          Logical operator is <code>EQUALS</code>.</p>\n          </li>\n          <li>\n
<p>\n                <code>Application Publisher</code> - The publisher of the application.\n          Logical
operator is <code>EQUALS</code>.</p>\n          </li>\n          <li>\n             <p>\n
<code>Application Version</code> - The version of the application.\n          Logical operator is
<code>EQUALS</code>.</p>\n          </li>\n          <li>\n             <p>\n                <code>Platform
Name</code> - The name of the platform.\n          Logical operator is <code>EQUALS</code>.</p>\n
</li>\n          <li>\n             <p>\n                <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
      </li>\n          <li>\n             <p>\n                <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>.  The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n          </p>\n          </li>\n          <li>\n             <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n          Logical operator is <code>NOT_EQUALS</code>.</p>\n          </li>\n
<li>\n             <p>\n                <code>License Included</code> - The type of license included.\n          Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n          Possible values are:
```

<code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-server-web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n
<p>The following filters and logical operators are supported when the resource type\n   is
<code>RDS</code>:</p>\n   \n   \n   <p>\n   <code>Engine Edition</code> -
The edition of the database engine.\n   Logical operator is <code>EQUALS</code>.\n   Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n   \n   \n   <p>\n   <code>License Pack</code> - The license
pack.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>data
guard</code> | \n   <code>diagnostic pack sqlt</code> | \n
          <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n   <code>olap</code>.</p>\n
\n   ",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#ProductInformationFilter": {
          "type": "structure",
          "members": {
            "ProductInformationFilterName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Filter name.</p>",
                "smithy.api#required": {}
              }
            },
            "ProductInformationFilterValue": {
              "target": "com.amazonaws.licensemanager#StringList",

          "traits": {
                "smithy.api#documentation": "<p>Filter value.</p>"
              }
            },
            "ProductInformationFilterComparator": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Logical operator.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes product information filters.</p>"
          }
        },

```
    "com.amazonaws.licensemanager#ProductInformationFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformationFilter"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformation"
      }
    },
    "com.amazonaws.licensemanager#ProvisionalConfiguration": {
      "type": "structure",
      "members": {
        "MaxTimeToLiveInMinutes": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#RateLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "RateLimitExceeded",
          "httpResponseCode": 429
        },
        "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 429
      }
    },
    "com.amazonaws.licensemanager#ReceivedMetadata": {
      "type": "structure",
      "members": {
```

```json
        "ReceivedStatus": {
          "target": "com.amazonaws.licensemanager#ReceivedStatus",
          "traits": {
            "smithy.api#documentation": "<p>Received status.</p>"
          }
        },
        "ReceivedStatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Received
status reason.</p>"
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReceivedStatus": {
      "type": "enum",
      "members": {
        "PENDING_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_WORKFLOW"
          }
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"PENDING_ACCEPT"
          }
        },
        "REJECTED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"
          }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
```

```
        "traits": {
          "smithy.api#enumValue": "ACTIVE"
        }
      },
      "FAILED_WORKFLOW": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "FAILED_WORKFLOW"
        }
      },
      "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DELETED"
        }
      },
      "DISABLED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DISABLED"
        }
      },
      "WORKFLOW_COMPLETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "WORKFLOW_COMPLETED"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#RedirectException": {
    "type": "structure",
    "members": {
      "Location": {
        "target": "com.amazonaws.licensemanager#Location",
        "traits": {
          "smithy.api#httpHeader": "Location"
        }
      },
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 308
    }
```

        },
      "com.amazonaws.licensemanager#RejectGrant": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#RejectGrantRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#RejectGrantResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#RejectGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },

```
          "traits": {
            "smithy.api#input": {}


    }
        },
      "com.amazonaws.licensemanager#RejectGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#RenewType":
{
        "type": "enum",
        "members": {
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "WEEKLY": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Weekly"
            }
          },
          "MONTHLY": {
```

```
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Monthly"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#ReportContext": {
        "type": "structure",
        "members": {
            "licenseConfigurationArns": {
                "target": "com.amazonaws.licensemanager#ArnList",
                "traits": {

        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportFrequency": {
        "type": "structure",
        "members": {
            "value": {
                "target": "com.amazonaws.licensemanager#Integer",
                "traits": {
                    "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
                }
            },
            "period": {
                "target": "com.amazonaws.licensemanager#ReportFrequencyType",

                "traits": {
                    "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportFrequencyType": {
```

```
          "type": "enum",
          "members": {
            "DAY": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "DAY"
              }
            },
            "WEEK": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "WEEK"
              }
            },
            "MONTH": {
              "target": "smithy.api#Unit",
              "traits":
{
                "smithy.api#enumValue": "MONTH"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#ReportGenerator": {
          "type": "structure",
          "members": {
            "ReportGeneratorName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>"
              }
            },
            "ReportType": {
              "target": "com.amazonaws.licensemanager#ReportTypeList",
              "traits": {
                "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
              }
            },
            "ReportContext": {
              "target": "com.amazonaws.licensemanager#ReportContext",
              "traits": {
                "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
              }
            },
            "ReportFrequency": {
              "target": "com.amazonaws.licensemanager#ReportFrequency",
              "traits": {
```

```
              "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
            }
          },
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
            }
          },
          "LastRunStatus": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
            }
          },
          "LastRunFailureReason":
{

            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
            }
          },
          "LastReportGenerationTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
            }
          },
          "ReportCreatorAccount": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",

      "traits": {
              "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
          },
          "S3Location": {
            "target": "com.amazonaws.licensemanager#S3Location",
            "traits": {
              "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
            }
```

```
        },
        "CreateTime": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the report was created.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
          }
        }

      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
    }
  },
  "com.amazonaws.licensemanager#ReportGeneratorList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ReportGenerator"
    }
  },
  "com.amazonaws.licensemanager#ReportGeneratorName": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "com.amazonaws.licensemanager#ReportType": {
    "type": "enum",
    "members": {
      "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
        }

      },
        "LICENSE_CONFIGURATION_USAGE_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationUsageReport"
```

```
            }
          }
        }
      },
      "com.amazonaws.licensemanager#ReportTypeList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#ReportType"
        }
      },
      "com.amazonaws.licensemanager#ResourceInventory": {
        "type": "structure",
        "members": {
          "ResourceId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits":
{
              "smithy.api#documentation": "<p>Type of resource.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "Platform": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Platform of the resource.</p>"
            }
          },
          "PlatformVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
            }
          },
          "ResourceOwningAccountId":
{
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
```

```
          "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about a resource.</p>"
    }
  }
},
"com.amazonaws.licensemanager#ResourceInventoryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanager#ResourceInventory"
  }
},
"com.amazonaws.licensemanager#ResourceLimitExceededException": {
  "type": "structure",
  "members": {
    "Message": {
      "target": "com.amazonaws.licensemanager#Message"
    }
  },
  "traits": {
    "aws.protocols#awsQueryError": {
      "code": "ResourceLimitExceeded",


      "httpResponseCode": 400
    },
    "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 400
  }
},
"com.amazonaws.licensemanager#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "Message": {
      "target": "com.amazonaws.licensemanager#Message"
    }
  },
  "traits": {
    "aws.protocols#awsQueryError": {
      "code": "InvalidResource.NotFound",
      "httpResponseCode": 400
    },
    "smithy.api#documentation": "<p>The resource cannot be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 400
  }
```

```
        },
        "com.amazonaws.licensemanager#ResourceType": {
            "type": "enum",
            "members": {
                "EC2_INSTANCE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "EC2_INSTANCE"
                    }
                },
                "EC2_HOST": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "EC2_HOST"
                    }
                },
                "EC2_AMI": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "EC2_AMI"
                    }
                },
                "RDS": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "RDS"
                    }
                },
                "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#S3Location": {
            "type": "structure",
            "members": {
                "bucket": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
                    }
                },
                "keyPrefix": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
```

```
                "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
    }
},
"com.amazonaws.licensemanager#ServerInternalException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
    },
    "traits": {
        "aws.protocols#awsQueryError": {
            "code": "InternalError",
            "httpResponseCode": 500
        },
        "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
    }
},
"com.amazonaws.licensemanager#SignedToken": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 4096
        }
    }
},
"com.amazonaws.licensemanager#StatusReasonMessage": {
    "type": "string",
    "traits":
    {
        "smithy.api#length": {
            "min": 0,
            "max": 400
        },
        "smithy.api#pattern": "^[\\s\\S]+$"
    }
},
"com.amazonaws.licensemanager#String": {
    "type": "string"
},
```

```
"com.amazonaws.licensemanager#StringList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#String"
    }
},
"com.amazonaws.licensemanager#Tag": {
    "type": "structure",
    "members": {
        "Key": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Tag key.</p>"
            }
        },
        "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation":
"<p>Tag value.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
    }
},
"com.amazonaws.licensemanager#TagKeyList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#String"
    }
},
"com.amazonaws.licensemanager#TagList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#Tag"
    }
},
"com.amazonaws.licensemanager#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#TagResourceResponse"
    },
    "errors": [
```

```
            {
               "target":
    "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
               "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
               "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
               "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
         }
      },
      "com.amazonaws.licensemanager#TagResourceRequest": {
         "type": "structure",
         "members": {
            "ResourceArn": {
               "target": "com.amazonaws.licensemanager#String",
               "traits":
    {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
    configuration.</p>",
                  "smithy.api#required": {}
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanager#TagList",
               "traits": {
                  "smithy.api#documentation": "<p>One or more tags.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanager#TagResourceResponse": {
         "type": "structure",
         "members": {},
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#TokenData": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
        "TokenType": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "ExpirationTime": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
format.</p>"
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
            "smithy.api#documentation": "<p>Data specified by the caller.</p>"
          }
        },
        "RoleArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
```

```
          "traits": {
              "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
          }
        }
      },
      "traits": {

        "smithy.api#documentation": "<p>Describes a token.</p>"
      }
    },
    "com.amazonaws.licensemanager#TokenList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#TokenData"
      }
    },
    "com.amazonaws.licensemanager#TokenString": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 4096
        },
        "smithy.api#pattern": "^\\S+$"
      }
    },
    "com.amazonaws.licensemanager#TokenType": {
      "type": "enum",
      "members": {
        "REFRESH_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REFRESH_TOKEN"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
{
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
```

again.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
      "com.amazonaws.licensemanager#UntagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#UntagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#UntagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
             },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
          }
        },
      "com.amazonaws.licensemanager#UntagResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanager#TagKeyList",

```
          "traits": {
            "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
      },
      "output": {

"target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseConfigurationStatus": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
            "traits": {
              "smithy.api#documentation": "<p>New status of the license configuration.</p>"
            }
          },
          "LicenseRules": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {

  "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
  configuration is licenseAffinityToHost.</p>"
            }
          },
          "LicenseCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
            }
          },
          "LicenseCountHardLimit": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
            }
          },
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
            }
          },
          "Description": {
```

```json
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>New description of the license configuration.</p>"
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>New product information.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        }
      },
      "traits":
{
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
    }
  },
  "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
    "type": "structure",
    "members": {
      "LicenseManagerReportGeneratorArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
          "smithy.api#required": {}
        }
      },
      "ReportGeneratorName": {
        "target": "com.amazonaws.licensemanager#ReportGeneratorName",
        "traits": {
          "smithy.api#documentation": "<p>Name of the report generator.</p>",
          "smithy.api#required": {}
        }
      },
      "Type": {
        "target": "com.amazonaws.licensemanager#ReportTypeList",
        "traits": {
          "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n        <ul>\n        <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n        </li>\n        <li>\n        <p>Resource report - Reports the tracked licenses and
```

resource consumption for a license configuration.</p>\n   \n   ",
                "smithy.api#required": {}
            }
        },
        "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
                "smithy.api#documentation": "<p>The report context.</p>",
                "smithy.api#required": {}
            }
        },
        "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits":
{

                "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                "smithy.api#required": {}
            }
        },
        "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientRequestToken",
            "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                "smithy.api#required": {}
            }
        },
        "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
        }
    }
},
    "traits": {
        "smithy.api#input": {}
    }
},
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
        "type": "operation",

```
"input": {
    "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
},
"output": {
    "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
},
"errors": [
    {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },

    {
        "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
    },
    {
        "target": "com.amazonaws.licensemanager#LicenseUsageException"
    },
    {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
    }
},

"com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
                "smithy.api#required": {}
            }
```

```
            },
            "AddLicenseSpecifications": {
              "target": "com.amazonaws.licensemanager#LicenseSpecifications",
              "traits": {
                "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
              }
            },
            "RemoveLicenseSpecifications": {
              "target": "com.amazonaws.licensemanager#LicenseSpecifications",
              "traits": {
                "smithy.api#documentation":
"<p>ARNs of the license configurations to remove.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#UpdateServiceSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
```

```
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
          }
        },
        "SnsTopicArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
          }
        },
        "OrganizationConfiguration": {
          "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
          }
        },
        "EnableCrossAccountsDiscovery": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

        }
      }
    },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {},
      "traits": {
```

```
              "smithy.api#output": {}
          }
      },
      "com.amazonaws.licensemanager#UsageOperation": {
          "type": "string",
          "traits": {
              "smithy.api#length": {
                  "min": 0,
                  "max": 50
              }
          }
      },
      "com.amazonaws.licensemanager#ValidationException": {
          "type": "structure",
          "members": {
              "Message": {
                  "target": "com.amazonaws.licensemanager#Message"
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

          "smithy.api#error": "client",
              "smithy.api#httpError": 400
          }
      }
  }
}
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)


// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }
```

```go
	result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*DeleteLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type DeleteLicenseInput struct {

	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// Current version
	of the license.
	//
	// This member is required.
	SourceVersion *string

	noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

	// Date when the license is deleted.
	DeletionDate *string

	// License status.
	Status types.LicenseDeletionStatus

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
	if err != nil {
```

```
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack,
options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
```

```
    err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
     return err
    }
    if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
     return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
    middleware.Before); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
    }

    func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
     return &awsmiddleware.RegisterServiceMetadata{
      Region:       region,
      ServiceID:     ServiceID,
      OperationName: "DeleteLicense",
     }
    }
```

1.32 aws-aws-sdk-go-v2-service-lambda 1.49.1

1.32.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
```

```go
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/grafana/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
// Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
	if params == nil {
		params = &AssociateLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
	if err != nil {
		return nil,
	err
	}

	out := result.(*AssociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type AssociateLicenseInput struct {

	// The type of license to associate with the workspace.
	//
	// This member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to associate the license with.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {
```

```go
// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err !=
nil {
 return err
 }
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
 }
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
 }

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
 }
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
```

```go
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack);
	err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack,
options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
```

```
    return &awsmiddleware.RegisterServiceMetadata{
      Region:        region,
      ServiceID:      ServiceID,
      OperationName: "AssociateLicense",
    }
}
```
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.

```yaml
name: License Scan

on: [pull_request]

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
```

```yaml
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
```

```go
	// This member is required.
	LicenseConsumptionToken *string

	// License beneficiary.
	Beneficiary
	*string

	noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
```

```
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return
    err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
```

```go
	return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckInLicense",
	}
}
```

```json
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
            "type": "structure",
            "members": {
                "Name": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
                    }
                },
                "Values": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                    "traits": {
                        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
                    }
                },
                "Operator": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
                    "traits": {
                        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
```

```
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
```

```
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n      Organizations integration has been enabled, and
 which Regions data will be aggregated from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays
 the aggregated data for Linux\n      subscriptions.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
          }
        },
        "InstanceID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
          }
        },
        "InstanceType": {
```

```
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance type of the resource.</p>"
          }

      },
      "AccountID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the instance.</p>"
        }
      },
      "Region": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
        }
      },
      "UsageOperation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
        }
      },
      "ProductCode": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
        "traits": {
          "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
        }
      },
      "LastUpdatedTime": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
        }
      },
```

```
    "SubscriptionName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
```

```
    "traits": {
        "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
            "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                "x-api-key",
                "authorization",
                "x-amz-date",

    "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
            ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
                "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
                    "documentation": "The
AWS region used to dispatch the request.",
                    "type": "String"
                },
                "UseDualStack": {
```

```
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

          "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {

          "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
```

```
            }
          ],
          "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
          "type": "error"
        },
        {
          "conditions": [],
          "type": "tree",
          "rules": [
            {

          "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
              "type": "error"
            },
            {
              "conditions": [],

      "endpoint": {
                "url": {
                  "ref": "Endpoint"
                },
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ]
        }
      ]
    },
    {
      "conditions": [],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {
```

```
                        "fn": "isSet",

          "argv": [
                          {
                            "ref": "Region"
                          }
                      ]
                  }
              ],
              "type": "tree",
              "rules": [
                {
                    "conditions": [
                        {
                            "fn": "aws.partition",
                            "argv": [
                              {
                                  "ref": "Region"
                              }
                            ],

"assign": "PartitionResult"
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                              {
                                  "fn": "booleanEquals",
                                  "argv": [
                                      {
                                          "ref": "UseFIPS"
                                      },
                                      true
                                  ]
                              },

              {
                                  "fn": "booleanEquals",
                                  "argv": [
                                      {
                                          "ref": "UseDualStack"
                                      },
                                      true
                                  ]
                              }
                            ],
```

```
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [
                          {

                      "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                              ]

}
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },

                      "supportsDualStack"
                              ]
                            }
                          ]
                        }
                      ],
                      "type": "tree",
                      "rules": [
                        {
                          "conditions": [],
                          "type": "tree",
                          "rules": [
                            {

                  "conditions": [],
                              "endpoint": {
```

```
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ]
                            }
                        ]

                    },
                    {
                        "conditions": [],
                        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                        "type": "error"
                    }
                ]
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {

            "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,

    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
```

```
                            "supportsFIPS"
                        ]
                    }
                ]
            }
        ],
        "type": "tree",

    "rules": [
        {
            "conditions": [],
            "type": "tree",
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                        "properties": {},
                        "headers": {}

                    },
                    "type": "endpoint"
                }
            ]
        }
    ]
},
        {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"
        }
    ]

},
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],
```

```
"type": "tree",
"rules": [
  {

    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ],
    "type": "tree",
    "rules": [
      {
        "conditions": [],
        "type": "tree",
        "rules": [
          {
            "conditions": [],

            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
              "properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ]
      }
    ]
  },

  {

    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
```

```json
                            ]
                          },
                          {
                            "conditions": [],
                            "type": "tree",
                            "rules": [
                              {
                                "conditions": [],
                                "endpoint": {
                                  "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
                                "type": "endpoint"
                              }
                            ]
                          }
                        ]
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "error": "Invalid Configuration: Missing Region",
                    "type": "error"
                  }
                ]
              }
            ]
          },
          "smithy.rules#endpointTests": {
            "testCases": [
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
```

```json
      "expect": {

"endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
         "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {

"url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
```

```
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
         "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint":

{

          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
```

```
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
           "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint":

{

            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
           "UseDualStack": false
        }
```

```
      },
      {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": true
          }
      },
      {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
              "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
              }
          },
          "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": false
          }
      },
      {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
              "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": true
          }
      },
      {

          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
              "endpoint": {
                  "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
              }
          },
          "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": false
```

```
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {

         "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
          }
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
          }
        },
        "params": {
          "Region": "us-isob-east-1",
```

```
        "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,

        "Endpoint": "https://example.com"
            }
        },
        {
```

                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }

        }
            ],
                "version": "1.0"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled LinuxSubscriptionsDiscovery"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
        "type": "structure",
        "members": {
            "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type":
"operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with
commercial Linux\n    subscriptions.</p>",
      "smithy.api#http": {

    "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
```

```
                    "outputToken": "NextToken",
                    "pageSize": "MaxResults",
                    "items": "Instances"
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
            "type": "structure",
            "members": {
                "Filters": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by
 the name of\n      <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n        <p>The valid names for this filter
are:</p>\n        <ul>\n          <li>\n            <p>\n                <code>AmiID</code>\n            </p>\n
</li>\n          <li>\n            <p>\n                <code>InstanceID</code>\n            </p>\n          </li>\n
<li>\n          <p>\n                <code>AccountID</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
 <code>InstanceType</code>\n            </p>\n          </li>\n        </ul>\n        <p>The valid Operators for this
filter are:</p>\n        <ul>\n          <li>\n            <p>\n                <code>contains</code>\n            </p>\n
</li>\n          <li>\n            <p>\n                <code>equals</code>\n            </p>\n          </li>\n          <li>\n
    <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
```

                    "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
                }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
                "type": "structure",
                "members": {
                    "Instances": {
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                        "traits": {
                            "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
                        }
                    },
                    "NextToken": {
                        "target": "smithy.api#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                        }
                    }
                }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
                "type": "operation",
                "input": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
                },
                "output": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
                },
                "errors": [
                    {
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                    }
                ],
                "traits": {
                    "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n   organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
                    "smithy.api#http": {
                        "code": 200,
                        "method": "POST",

```
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n     partially match, or don't
match a certain subscription's name.</p>\n        <p>The valid names for this filter are:</p>\n        <ul>\n
<li>\n           <p>\n               <code>Subscription</code>\n            </p>\n         </li>\n        </ul>\n
<p>The valid Operators for this filter are:</p>\n

       <ul>\n         <li>\n          <p>\n               <code>contains</code>\n           </p>\n         </li>\n
<li>\n          <p>\n              <code>equals</code>\n           </p>\n         </li>\n         <li>\n          <p>\n
         <code>Notequal</code>\n            </p>\n         </li>\n        </ul>"
          }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>",
          "smithy.api#length": {

           "min": 1,
            "max": 16384
          }
        }
      }
    },
    "traits": {
```

```
                "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
            "type": "structure",
            "members": {
                "Subscriptions": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                    "traits": {
                        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "EQUAL",
                        "value": "Equal",
                        "documentation": "Equal operator"
                    },
                    {
                        "name": "NOT_EQUAL",
                        "value": "NotEqual",
                        "documentation": "Not equal operator"
                    },
                    {
                        "name": "CONTAINS",
                        "value": "Contains",
                        "documentation": "Contains operator"
                    }
                ],
                "smithy.api#length": {
                    "min": 1,
                    "max": 20
                }
            }
        },
```

```machine_data
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [

{
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
```

```
                }
              ]
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
            "type": "list",
            "member": {
              "target": "smithy.api#String",
              "traits":
{
                  "smithy.api#length": {
                    "min": 1,
                    "max": 100
                  }
                }
            },
            "traits": {
              "smithy.api#length": {
                "min": 1,
                "max": 100
              },
              "smithy.api#uniqueItems": {}
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
            "type": "map",
            "key": {
              "target": "smithy.api#String"
            },
            "value": {
              "target": "smithy.api#String"
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
            "type": "structure",
            "members": {
              "Name": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The name of the subscription.</p>"

              }
            },
            "Type": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n     Subscription model (BYOS), or from the Amazon Web Services
```

Marketplace. Certain subscriptions may use licensing from the\n   Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
                    }
                },
                "InstanceCount": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
                    "traits": {
                        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
                "smithy.api#error": "client"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
            },

            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
                },
                {

```
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/subscription/UpdateServiceSettings"
                },
                "smithy.api#idempotent": {}
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
            "type": "structure",
            "members": {
                "LinuxSubscriptionsDiscovery": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                    "traits": {
                        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                        "smithy.api#required": {}
                    }
                },
                "LinuxSubscriptionsDiscoverySettings": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                    "traits": {
                        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>",

    "smithy.api#required": {}
                    }
                },
                "AllowUpdate": {
                    "target": "smithy.api#Boolean",
                    "traits": {
                        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n     allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
```

```
"type": "structure",
"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    }
  },

  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n     integration has been enabled, and which Regions data will be aggregated
from.</p>"
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n     subscriptions.</p>"
    }
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
```

```
                "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
                "smithy.api#error": "client"
            }
        }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //   - Beneficiary
 //   - ProductSKU
 //   - Fingerprint
 //
    - Status
 Filters []types.Filter
```

```go
    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if
    err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
```

```go
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
    return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
    err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
    err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
```

```go
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:      region,
		ServiceID:
			ServiceID,
		OperationName: "ListLicenses",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/grafana/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
	if params == nil {
		params = &DisassociateLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*DisassociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}
```

```go
type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    //
    This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
    options, "DisassociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
```

```go
	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
```

```go
    return err
   }
   if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
   }
   if err = addResponseErrorMiddleware(stack); err != nil {
    return err
   }
   if err = addRequestResponseLogging(stack, options); err != nil {
    return err
   }
   if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
   }
   return nil
  }

  func newServiceMetadataMiddleware_opDisassociateLicense(region string)
  *awsmiddleware.RegisterServiceMetadata
   {
   return &awsmiddleware.RegisterServiceMetadata{
    Region:       region,
    ServiceID:    ServiceID,
    OperationName: "DisassociateLicense",
   }
  }
  // Code generated by smithy-go-codegen DO NOT EDIT.

  package licensemanager

  import (
   "context"
   "fmt"
   awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
   "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
   "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
   "github.com/aws/smithy-go/middleware"
   smithyhttp "github.com/aws/smithy-go/transport/http"
  )

  // Checks out the specified license. If the account that created the license is
  // the same that is performing the check out, you must specify the account as the
  // beneficiary.
  func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
  (*CheckoutLicenseOutput, error) {
   if params == nil {
    params = &CheckoutLicenseInput{}
   }
```

```go
	result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*CheckoutLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type
	CheckoutLicenseInput struct {

	// Checkout type.
	//
	// This member is required.
	CheckoutType types.CheckoutType

	// Unique, case-sensitive identifier that you provide to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// License entitlements.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Key fingerprint identifying the license.
	//
	// This member is required.
	KeyFingerprint *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// License beneficiary.
	Beneficiary *string

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
```

```go
}

type CheckoutLicenseOutput struct {

	// Checkout type.
	CheckoutType types.CheckoutType

	// Allowed license entitlements.
	EntitlementsAllowed []types.EntitlementData

	// Date and time at which the license checkout expires.
	Expiration *string

	// Date and time at which the license checkout is issued.
	IssuedAt *string

	// Amazon Resource
	Name (ARN) of the checkout license.
	LicenseArn *string

	// License consumption token.
	LicenseConsumptionToken *string

	// Node ID.
	NodeId *string

	// Signed token.
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
```

```
	return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil
{
	return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
	return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
	return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
	return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
	return err
}
if err = addClientUserAgent(stack, options); err != nil {
	return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
	return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack);
err != nil {
	return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
	return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
```

```go
    middleware.Before); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
     Region:       region,
     ServiceID:    ServiceID,
     OperationName: "CheckoutLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
```

```go
	}

	result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*CheckoutBorrowLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type CheckoutBorrowLicenseInput struct {

	// Unique, case-sensitive identifier
	that you provide to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// Digital signature method. The possible value is JSON Web Signature (JWS)
	// algorithm PS384. For more information, see RFC 7518 Digital Signature with
	// RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
	//
	// This member is required.
	DigitalSignatureMethod types.DigitalSignatureMethod

	// License entitlements. Partial checkouts are not supported.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Amazon Resource Name (ARN) of the license. The license must use the borrow
	// consumption configuration.
	//
	// This member is required.
	LicenseArn *string

	// Information about constraints.
	CheckoutMetadata []types.Metadata

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}
```

```go
type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
```

```go
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
```

```go
 middleware.Before); err != nil {
  return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
  return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
  return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
  }
  return nil
 }

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
  }
 }
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
```

```go
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide
    to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer
```

```go
	// License name.
	//
	// This member is required.
	LicenseName *string

	// Product name.
	//
	// This member is required.
	ProductName *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// Date and time range during which the license is valid, in ISO8601-UTC format.
	//
	// This member is required.
	Validity
	*types.DatetimeRange

	// Information about the license.
	LicenseMetadata []types.Metadata

	noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
```

```
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add
protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
```

```go
    err != nil {
     return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
     return err
    }
    if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
     return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
    middleware.Before); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
    }

    func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
     return &awsmiddleware.RegisterServiceMetadata{
      Region:       region,
      ServiceID:    ServiceID,
      OperationName:
     "CreateLicense",
     }
    }
    // Code generated by smithy-go-codegen DO NOT EDIT.

    package licensemanager

    import (
     "context"
     "fmt"
     awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
     "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
```

```go
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
    c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version
    *string

    noSmithyDocumentSerde
}

type GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}
```

```go
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return
err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack);
 err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
 params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
 return nil, err
 }
```

```go
	out := result.(*ListReceivedLicensesOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type ListReceivedLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//
	//   - ProductSKU
	//   - Status
	//   - Fingerprint
	//   - IssuerName
	//   - Beneficiary
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

	// Received license details.
	Licenses []types.GrantedLicense

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
```

```go
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
	middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack);
	err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
```

```
  }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
  }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
  }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
  }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
  }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
  }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
  }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
  }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
  }
}
```

                    Apache License
                Version 2.0, January 2004
              http://www.apache.org/licenses/

 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

  1. Definitions.

    "License" shall mean the terms and conditions for use, reproduction,
    and distribution as defined by Sections 1 through 9 of this document.

    "Licensor" shall mean the copyright owner or entity authorized by
    the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise

designed in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
   distribution, then any Derivative Works that You distribute must
   include a readable copy of the attribution notices contained
   within such NOTICE file, excluding
those notices that do not
   pertain to any part of the Derivative Works, in at least one
   of the following places: within a NOTICE text file distributed
   as part of the Derivative Works; within the Source form or
   documentation, if provided along with the Derivative Works; or,
   within a display generated by the Derivative Works, if and
   wherever such third-party notices normally appear. The contents
   of the NOTICE file are for informational purposes only and
   do not modify the License. You may add Your own attribution
   notices within Derivative Works that You distribute, alongside
   or as an addendum to the NOTICE text from the Work, provided
   that such additional attribution notices cannot be construed
   as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions

 for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

 5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

 6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the
content of the NOTICE file.

 7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct,
indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
    incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

```json
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
         },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n        <note>\n          <p>Your estimated bill for charges on the number of users and related
costs will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
    monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/AssociateUser"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the
```

EC2 instance, which provides user-based subscriptions.</p>",

```
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n     resource.</p>",
            "smithy.api#error": "server"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
```

```json
{
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
```

```
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
```

```
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }

    },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
```

```
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      }
    },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },

  "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used
to return more specific results from a describe\n      operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n      attributes, or IDs.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n      settings such as the subnets to provision VPC endpoints.</p>",
            "smithy.api#required":
{}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
```

```
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status
of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
```

```
                "smithy.api#required": { }
              }
            },
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
              }

    }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": { }
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": { }
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
```

```
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
          "smithy.api#required": {}
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the user.</p>"
        }
      },
      "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
      },
      "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        }
      }

    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
```

            },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
                "smithy.api#error": "server"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
            "type": "service",

    "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
            },

},
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"

```
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
      },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
          "x-amzn-trace-id",
          "content-length",
          "x-api-key",
          "authorization",
          "x-amz-date",
          "x-amz-security-token",
          "Access-Control-Allow-Headers",
          "Access-Control-Allow-Methods",
          "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
          "x-amzn-errortype",
          "x-amzn-requestid",
          "x-amzn-trace-id"
        ]
      },
      "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
      "smithy.api#title": "AWS License Manager User Subscriptions",
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
          "Region": {
            "builtIn": "AWS::Region",
            "required": false,
```

```
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack
endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "Boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "String"
    }

},
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
```

```
                },

              true
              ]
            }
          ],
          "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
          "type": "error"
        },
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ],

  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
          "type": "error"
        },
        {
          "conditions": [],
          "endpoint": {
            "url": {
              "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [

  {
              "ref": "Region"
            }
          ]
```

```
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "aws.partition",
                                "argv": [
                                    {
                                        "ref": "Region"
                                    }
                                ],
                                "assign": "PartitionResult"
                            }
                        ],
                        "type": "tree",

"rules": [
                                {
                                    "conditions": [
                                        {
                                            "fn": "booleanEquals",
                                            "argv": [
                                                {
                                                    "ref": "UseFIPS"
                                                },
                                                true
                                            ]
                                        },
                                        {
                                            "fn": "booleanEquals",
                                            "argv": [
                                                {
                                                    "ref": "UseDualStack"

                                                },
                                                true
                                            ]
                                        }
                                    ],
                                    "type": "tree",
                                    "rules": [
                                        {
                                            "conditions": [
                                                {
                                                    "fn": "booleanEquals",
                                                    "argv": [
```

```
                                      true,
                                      {
                                        "fn": "getAttr",

      "argv": [
                                              {
                                                "ref": "PartitionResult"
                                              },
                                              "supportsFIPS"
                                          ]
                                      }
                                  ]
                              },
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  true,
                                  {

            "fn": "getAttr",
                                      "argv": [
                                        {
                                          "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                      ]
                                  }
                                ]
                              }
                          ],
                          "type": "tree",
                          "rules": [
                            {

        "conditions": [],
                                "endpoint": {
                                  "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
                                "type": "endpoint"
                            }
                          ]
                        },
                        {
                          "conditions": [],
                          "error": "FIPS and DualStack are enabled, but this partition does not support one or
```

both",
```
                              "type": "error"
                            }
                          ]
                        },
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                {
                                  "ref": "UseFIPS"
                                },
                                true
                              ]
                            }
                          ],
                          "type": "tree",

            "rules": [
                            {
                              "conditions": [
                                {
                                  "fn": "booleanEquals",
                                  "argv": [
                                    true,
                                    {
                                      "fn": "getAttr",
                                      "argv": [
                                        {
                                          "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"

                                      ]
                                    }
                                  ]
                                }
                              ],
                              "type": "tree",
                              "rules": [
                                {
                                  "conditions": [],
                                  "endpoint": {
                                    "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
```

```
                },
                                "type": "endpoint"
                            }
                        ]
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled but this partition does not support FIPS",
                        "type": "error"
                    }
                ]
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {

                                    "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
```

```
                                {

                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ]
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does
not support DualStack",
                            "type": "error"
                        }
                    ]
                },
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        }
    ]
},
{
    "conditions":
[],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
          },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
```

```
                            "UseDualStack": false
                        }
                    },
                    {
                        "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                            }
                        },
                        "params": {
                            "Region": "cn-north-1",
                            "UseFIPS": true,
                            "UseDualStack": true
                        }
                    },
                    {
                        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                            }
                        },

            "params": {
                            "Region": "cn-north-1",
                            "UseFIPS": true,
                            "UseDualStack": false
                        }
                    },
                    {
                        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                            }
                        },
                        "params": {
                            "Region": "cn-north-1",
                            "UseFIPS": false,
                            "UseDualStack": true
                        }
                    },
                    {
                        "documentation": "For region cn-north-1
```

with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                  }
                },

      "params": {
                  "Region": "us-gov-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-gov-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                  }
                },
                "params": {

          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
          }
        },

  "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect":

{
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "error": "DualStack is enabled but this partition does not support DualStack"

```
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack":
true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          }

    },
          {
```

"documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
           "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {

"documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,

```
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "Missing region",
            "expect": {
              "error": "Invalid Configuration: Missing Region"
            }
          }
        ],
        "version": "1.0"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
      {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
         },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/ListIdentityProviders"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "IdentityProviderSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
```

```
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
          "type": "structure",
          "members": {
            "IdentityProviderSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
                "smithy.api#required": {}
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
```

```json
        "type":
"structure",
        "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
```

```
      "traits": {
        "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListProductSubscriptions"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "ProductUserSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {

  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n    more sets of key-value pairs that you specify.</p>"
          }
        },
```

```
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },

        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
          },
          "smithy.api#paginated":
{
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
```

```
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for
 the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
          "type": "structure",
          "members": {

            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
```

```
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }

    },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for a user.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
          }
        },
        "SubscriptionEndDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
```

```
      "traits": {
        "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/RegisterIdentityProvider"
        },
```

```
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n      subnets to provision VPC endpoints.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
          "type": "structure",
          "members": {
            "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
        "type": "string",

  "traits": {
          "smithy.api#length": {
            "min": 5,
            "max": 200
          },
          "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
```

```
                    "traits": {
                        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n      your VPC and the VPC endpoint for activation servers.</p>",
                        "smithy.api#required": {}
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n      subnets to provision VPC endpoints, and the security group ID that
 is associated with the VPC\n      endpoints. The security group should permit inbound TCP port 1688
communication from resources\n      in the VPC.</p>"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n      <note>\n         <p>Your estimated bill for charges on the number of users and related costs
```

will take 48\n   hours to appear for billing periods that haven't closed (marked as Pending billing status) in Amazon Web Services Billing.
 For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n   monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p>\n   </note>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StartProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {

```json
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/StopProductSubscription"
```

```
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required":
{}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation":
"<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {

          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity\n    provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "UpdateSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n    update any combination of settings in a single operation such as the:</p>\n
<ul>\n      <li>\n          <p>Subnets which you want to add to provision VPC endpoints.</p>\n      </li>\n
<li>\n          <p>Subnets which you want to remove the VPC endpoints from.</p>\n      </li>\n
<li>\n          <p>Security group ID which permits traffic to the VPC endpoints.</p>\n      </li>\n    </ul>",
            "smithy.api#required": {}
          }
        }
      }
    },
```

```
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {

            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will create a VPC endpoint for products that\n     require connectivity to activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID
 that allows inbound TCP port 1688 communication between resources in\n     your VPC and the VPC endpoints for activation servers.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as\n     the subnets to provision VPC endpoints.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
  }
}
{
    "smithy": "2.0",
    "metadata": {
      "suppressions": [
        {
          "id": "HttpMethodSemantics",
          "namespace": "*"
        },
        {
          "id": "HttpResponseCodeSemantics",
          "namespace": "*"
        },
        {
          "id": "PaginatedTrait",
          "namespace": "*"
        },
        {
          "id": "HttpHeaderTrait",
          "namespace": "*"
        },
        {
          "id": "HttpUriConflict",
          "namespace": "*"
        },
        {
          "id": "Service",
          "namespace": "*"
        }
      ]
    },
    "shapes": {
      "com.amazonaws.licensemanager#AWSLicenseManager": {
        "type": "service",
        "version": "2018-08-01",
        "operations": [
          {
            "target": "com.amazonaws.licensemanager#AcceptGrant"
          },
```

```
          {
                  "target": "com.amazonaws.licensemanager#CheckInLicense"
          },
          {
                  "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
          },
          {
                  "target": "com.amazonaws.licensemanager#CheckoutLicense"
          },
          {
                  "target": "com.amazonaws.licensemanager#CreateGrant"
          },
          {
                  "target": "com.amazonaws.licensemanager#CreateGrantVersion"
          },
          {
                  "target": "com.amazonaws.licensemanager#CreateLicense"
          },
          {
                  "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
          },
          {
                  "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
          },
          {
                  "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
           },
          {
                  "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
          },
          {
                  "target": "com.amazonaws.licensemanager#CreateToken"
          },
          {
                  "target": "com.amazonaws.licensemanager#DeleteGrant"
          },
          {
                  "target": "com.amazonaws.licensemanager#DeleteLicense"
          },
          {
                  "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
          },
          {
                  "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
          },
          {
                  "target": "com.amazonaws.licensemanager#DeleteToken"
```

          },
          {
            "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"

          },
          {
            "target": "com.amazonaws.licensemanager#GetAccessToken"
          },
          {
            "target": "com.amazonaws.licensemanager#GetGrant"
          },
          {
            "target": "com.amazonaws.licensemanager#GetLicense"
          },
          {
            "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
          },
          {
            "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
          },
          {
            "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
          },
          {
            "target": "com.amazonaws.licensemanager#GetLicenseUsage"
          },
          {
            "target": "com.amazonaws.licensemanager#GetServiceSettings"
          },
          {
            "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
           },
          {
            "target": "com.amazonaws.licensemanager#ListDistributedGrants"
          },
          {
            "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
          },
          {
            "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
          },
          {
            "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
          },
          {
            "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
          },
          {

```
      "target": "com.amazonaws.licensemanager#ListLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
    },
    {

      "target": "com.amazonaws.licensemanager#ListLicenseVersions"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListResourceInventory"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTokens"
    },
    {
      "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#RejectGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
    },
```

```
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager",
        "arnNamespace": "license-manager",
        "cloudFormationName": "LicenseManager",
        "cloudTrailEventSource": "licensemanager.amazonaws.com",
        "endpointPrefix": "license-manager"
      },
      "aws.auth#sigv4": {
        "name": "license-manager"
      },
      "aws.protocols#awsJson1_1": {},
      "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n      Amazon Web Services accounts and on-premises servers.</p>",
      "smithy.api#title": "AWS License Manager",
      "smithy.api#xmlNamespace": {
        "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
      },
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
          "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
          "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send
```

this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

```json
                "type": "Boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "String"
              }
            },
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Endpoint"
                      }
                    ]

                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                  },
                  {

      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
```

```
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [],
        "endpoint": {
          "url": {
            "ref": "Endpoint"
          },
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ]
  },
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "type": "tree",
    "rules": [
      {
        "conditions": [
          {
            "fn": "aws.partition",
            "argv": [
              {
                "ref": "Region"
              }
            ],
            "assign": "PartitionResult"
          }
        ],
        "type": "tree",
```

"rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref":
"UseFIPS"
                        },
                        true
                    ]
                },
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {

                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },

    "supportsDualStack"

                                    ]
                                }
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}

                            },
                                "type": "endpoint"
                        }
                    ]
                },
                {
                    "conditions": [],
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                    "type": "error"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",

            "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],

```
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  true,

{
                    "fn": "getAttr",
                    "argv": [
                      {
                        "ref": "PartitionResult"
                      },
                      "supportsFIPS"
                    ]
                  }
                ]
              }
            ],
            "type": "tree",
            "rules": [

{
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ]
          },
          {
            "conditions": [],
            "error": "FIPS is enabled but this
partition does not support FIPS",
            "type": "error"
          }
        ]
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
```

```
                    {
                       "ref": "UseDualStack"
                    },
                    true
                 ]
              }
           ],
           "type": "tree",
           "rules": [
              {
                 "conditions": [
                    {
                       "fn": "booleanEquals",
                       "argv": [
                          true,
                          {
                             "fn": "getAttr",
                             "argv": [
                                {
                                   "ref": "PartitionResult"
                                },

      "supportsDualStack"

                             ]
                          }
                       ]
                    }
                 ],
                 "type": "tree",
                 "rules": [
                    {
                       "conditions": [],
                       "endpoint": {
                          "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
                          "properties": {},
                          "headers": {}

                       },
                       "type": "endpoint"
                    }
                 ]
              },
              {
                 "conditions": [],
                 "error": "DualStack is enabled but this partition does not support DualStack",
                 "type": "error"
              }
```

```
                            ]
                        },
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ]
                }
            ]
        },
        {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint":
{
                    "url": "https://license-manager.af-south-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "af-south-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.ap-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "ap-east-1",
                "UseFIPS": false,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-3.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }

        },
        {
          "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-south-1.amazonaws.com"
```

```
            }
          },
          "params": {
            "Region": "ap-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-1.amazonaws.com"

}
          },
          "params": {
            "Region": "ap-southeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

"documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-3.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }
```

```
        },
        {
          "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ca-central-1.amazonaws.com"
            }
          },

         "params": {
            "Region": "ca-central-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-central-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-central-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-north-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-south-1.amazonaws.com"
            }
```

```
                },
                "params": {
                    "Region":
"eu-south-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.eu-west-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "eu-west-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
                "expect": {

                    "endpoint": {
                        "url": "https://license-manager.eu-west-2.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "eu-west-2",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.eu-west-3.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "eu-west-3",
                    "UseFIPS": false,

"UseDualStack": false
                }
```

```
          },
          {
            "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.me-south-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "me-south-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.sa-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "sa-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }

    },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-east-1.amazonaws.com"
              }
            },
```

```
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-east-2.amazonaws.com"
            }

          },
          "params": {
            "Region": "us-east-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-east-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-2",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region
us-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-west-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
```

```
        "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-west-1.amazonaws.com"
          }
        },
        "params": {

    "Region": "us-west-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-west-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-west-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

"expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-west-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-west-2",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-east-1.api.aws"
          }
        },
        "params": {
```

```
                    "Region": "us-east-1",
                    "UseFIPS": true,

                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-northwest-1",
                    "UseFIPS": false,
                    "UseDualStack": false

}
            },
            {
```

```
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },

    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-gov-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-west-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-west-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
```

```json
                    "url": "https://license-manager-fips.us-gov-east-1.api.aws"
                  }
                },

        "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager.us-gov-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
        disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                  "expect": {
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                     "UseDualStack": true
                  }
                },
```

```
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "error": "DualStack is enabled
but this partition does not support DualStack"
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},

{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": true
```

```
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
                  }
                },

        "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "error": "DualStack is enabled but this partition does not support DualStack"
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

      "expect": {
                    "endpoint": {
                      "url": "https://example.com"
                    }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
        disabled",
                "expect": {
                  "endpoint": {
                      "url": "https://example.com"
```

```
          }
        },
        "params": {
          "UseFIPS": false,

  "UseDualStack": false,
          "Endpoint": "https://example.com"
        }
      },
      {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {
          "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false,
          "Endpoint": "https://example.com"
        }
      },
      {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
          "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true,
          "Endpoint": "https://example.com"
        }
      },
      {
        "documentation": "Missing region",
        "expect": {
          "error": "Invalid Configuration: Missing Region"
        }
      }
    ],
    "version": "1.0"
  }
}
},
"com.amazonaws.licensemanager#AcceptGrant": {
  "type": "operation",
  "input": {
```

```
          "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

},
      "output": {
        "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }

    }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

```
        },
        "com.amazonaws.licensemanager#AcceptGrantResponse": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Grant
ARN.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanager#GrantStatus",
              "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#AccessDeniedException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {

  "code": "ServiceAccessDenied",
              "httpResponseCode": 401
            },
            "smithy.api#documentation": "<p>Access to resource denied.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 401
          }
        },
        "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
          "type": "enum",
```

```
    "members": {
      "DISTRIBUTED_GRANTS_ONLY": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
        }
      },
      "ALL_GRANTS_PERMITTED_BY_ISSUER": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#AllowedOperation": {

    "type": "enum",
      "members": {
        "CREATE_GRANT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateGrant"
          }
        },
        "CHECKOUT_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutLicense"
          }
        },
        "CHECKOUT_BORROW_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutBorrowLicense"
          }
        },
        "CHECK_IN_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckInLicense"
          }
        },
        "EXTEND_CONSUMPTION_LICENSE":
{

          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ExtendConsumptionLicense"
```

```
          }
        },
        "LIST_PURCHASED_LICENSES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ListPurchasedLicenses"
          }
        },
        "CREATE_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateToken"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#AllowedOperationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#AllowedOperation"
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max":
7
        }
      }
    },
    "com.amazonaws.licensemanager#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 2048
        },
        "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
      }
    },
    "com.amazonaws.licensemanager#ArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
      }
    },
    "com.amazonaws.licensemanager#AuthorizationException": {
      "type": "structure",
```

```
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError":
{
          "code": "AuthorizationFailure",
          "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n        policy associated with this account.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 403
      }
    },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
      "type": "structure",
      "members": {
        "LastRunTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes automated discovery.</p>"
      }
    },

  "com.amazonaws.licensemanager#Boolean": {
      "type": "boolean",
      "traits": {
        "smithy.api#default": false
      }
    },
    "com.amazonaws.licensemanager#BorrowConfiguration": {
      "type": "structure",
      "members": {
        "AllowEarlyCheckIn": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
            "smithy.api#required": {}
          }
        },
```

```
          "MaxTimeToLiveInMinutes": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
              "smithy.api#required": {}
            }
          }

        },
        "traits": {
          "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#BoxBoolean": {
        "type": "boolean"
      },
      "com.amazonaws.licensemanager#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanager#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanager#CheckInLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"

        },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
```

                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              },
              {
                "target": "com.amazonaws.licensemanager#ValidationException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
            }
        },
        "com.amazonaws.licensemanager#CheckInLicenseRequest":
{
            "type": "structure",
            "members": {
              "LicenseConsumptionToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>License consumption token.</p>",
                  "smithy.api#required": {}
                }
              },
              "Beneficiary": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>License beneficiary.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#CheckInLicenseResponse": {
            "type": "structure",
            "members": {},
            "traits": {
              "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"

    },
    "output": {
      "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
      },
      {
        "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
    }
  },
  "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
    "type": "structure",
    "members":

```json
{
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
            "smithy.api#required": {}
          }
        },
        "DigitalSignatureMethod": {
          "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
          "traits": {
            "smithy.api#documentation": "<p>Digital signature method. The possible
 value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
            "smithy.api#required": {}
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>"
            }
          },
          "EntitlementsAllowed":
{
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          },
          "SignedToken": {
            "target": "com.amazonaws.licensemanager#SignedToken",
            "traits": {
              "smithy.api#documentation": "<p>Signed token.</p>"
            }
          },
          "IssuedAt": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
            }
          },
          "Expiration": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
            }
```

```
            },
        "CheckoutMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
                "smithy.api#documentation": "<p>Information about constraints.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#CheckoutLicense": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target":
"com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Checks out the specified license.</p>\n         <note>\n            <p>If the
account that created the license is the same that is performing the check out, you must\n            specify the account
as the beneficiary.</p>\n         </note>"
        }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
        "type": "structure",
        "members": {
            "ProductSKU": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Product
SKU.</p>",
                    "smithy.api#required": {}
                }
            },
            "CheckoutType": {
                "target": "com.amazonaws.licensemanager#CheckoutType",
                "traits": {
                    "smithy.api#documentation": "<p>Checkout type.</p>",
                    "smithy.api#required": {}
                }
            },
            "KeyFingerprint": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
                    "smithy.api#required": {}
                }
            },
            "Entitlements": {
                "target": "com.amazonaws.licensemanager#EntitlementDataList",
                "traits": {
                    "smithy.api#documentation": "<p>License entitlements.</p>",

"smithy.api#required": {}
                }
            },
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
```

idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            },
            "Beneficiary": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License beneficiary.</p>"
                }
            },
            "NodeId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Node ID.</p>"
                }
            }
        },
        "traits": {

        "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
        "type": "structure",
        "members": {
            "CheckoutType": {
                "target": "com.amazonaws.licensemanager#CheckoutType",
                "traits": {
                    "smithy.api#documentation": "<p>Checkout type.</p>"
                }
            },
            "LicenseConsumptionToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License consumption token.</p>"
                }
            },
            "EntitlementsAllowed": {
                "target": "com.amazonaws.licensemanager#EntitlementDataList",
                "traits": {
                    "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
                }
            },
            "SignedToken": {
                "target": "com.amazonaws.licensemanager#SignedToken",
                "traits": {
                    "smithy.api#documentation": "<p>Signed token.</p>"
                }
```

```
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "IssuedAt": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
          }
        },
        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutType": {
      "type": "enum",
      "members": {
        "PROVISIONAL": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PROVISIONAL"
          }
        },
        "PERPETUAL": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PERPETUAL"
          }
        }
      }
    },
```

```
"com.amazonaws.licensemanager#ClientRequestToken": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 36
        }
    }
},
"com.amazonaws.licensemanager#ClientToken": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 2048
        },
        "smithy.api#pattern": "^\\S+$"
    }
},
"com.amazonaws.licensemanager#ConflictException": {
    "type": "structure",
    "members": {
        "Message": {
            "target": "com.amazonaws.licensemanager#Message"
        }
    },
    "traits": {
        "aws.protocols#awsQueryError": {
            "code": "ConflictException",
            "httpResponseCode":
409
        },
        "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 409
    }
},
"com.amazonaws.licensemanager#ConsumedLicenseSummary": {
    "type": "structure",
    "members": {
        "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
                "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
            }
        },
        "ConsumedLicenses": {
```

```
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
            }
          }
        }
      },

  "traits": {
          "smithy.api#documentation": "<p>Details about license consumption.</p>"
        }
      },
    "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
      }
    },
    "com.amazonaws.licensemanager#ConsumptionConfiguration": {
      "type": "structure",
      "members": {
        "RenewType": {
          "target": "com.amazonaws.licensemanager#RenewType",
          "traits": {
            "smithy.api#documentation": "<p>Renewal frequency.</p>"
          }
        },
        "ProvisionalConfiguration": {
          "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
          }
        },
        "BorrowConfiguration": {
          "target": "com.amazonaws.licensemanager#BorrowConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantRequest"
      },
```

```
"output": {
  "target": "com.amazonaws.licensemanager#CreateGrantResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanager#AccessDeniedException"
  },
  {
    "target":
"com.amazonaws.licensemanager#AuthorizationException"
  },
  {
    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
  },
  {
    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
  },
  {
    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
  },
  {
    "target": "com.amazonaws.licensemanager#ServerInternalException"
  },
  {
    "target": "com.amazonaws.licensemanager#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
  }
},
"com.amazonaws.licensemanager#CreateGrantRequest": {
  "type": "structure",
  "members": {
    "ClientToken": {
      "target": "com.amazonaws.licensemanager#ClientToken",
      "traits": {
        "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
        "smithy.api#required": {}
      }
    },
    "GrantName": {
      "target": "com.amazonaws.licensemanager#String",
```

```
              "traits": {
                "smithy.api#documentation": "<p>Grant name.</p>",
                "smithy.api#required": {}
              }

            }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Principals": {
          "target": "com.amazonaws.licensemanager#PrincipalArnList",
          "traits": {
            "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n      (ARN):</p>\n      <ul>\n         <li>\n            <p>An Amazon Web Services
account, which includes only the account specified.</p>\n         </li>\n      </ul>\n      <ul>\n         <li>\n
   <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n         </li>\n      </ul>\n
<ul>\n
      <li>\n            <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n      </ul>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },

    "com.amazonaws.licensemanager#CreateGrantResponse": {
      "type": "structure",
```

```
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersion":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        }
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
      "type": "structure",
      "members": {
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
          }
        },
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant name.</p>"
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "StatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Grant status reason.</p>"
          }
        },
        "SourceVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Current version of the grant.</p>"
          }
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
            "smithy.api#documentation": "<p>The options specified for the grant.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>New version of the grant.</p>"
          }
        }
```

```
        },
        "traits":
{
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicense": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a license.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
 all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of
licenses purchased and used.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the license configuration.</p>"
            }
          },
          "LicenseCountingType": {
            "target": "com.amazonaws.licensemanager#LicenseCountingType",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
```
Exceeding a hard limit\n   blocks the launch of new instances.</p>"
```
          }
        },
        "LicenseRules": {
          "target":
 "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
```
#allowedTenancy=EC2-DedicatedHost). The available rules \n   vary by dimension, as follows.</p>\n
\n   \n   <p>\n   <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
  <code>licenseAffinityToHost</code> |\n   <code>maximumCores</code> |
<code>minimumCores</code>\n   </p>\n   \n   \n   <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n   <code>maximumCores</code> |
<code>minimumCores</code> |\n   <code>maximumSockets</code> | <code>minimumSockets</code> |\n
    <code>maximumVcpus</code> | <code>minimumVcpus</code>\n   </p>\n   \n
\n
  <p>\n   <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n   <code>maximumSockets</code> |
<code>minimumSockets</code>\n   </p>\n   \n   \n   <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n   <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n   </p>\n   \n   \n   <p>The unit for
<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for
<code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n
<code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n
<code>True</code> and <code>False</code>.</p>"
```
          }
        },

    "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
```

```
                }
              },
              "DisassociateWhenNotFound": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                  "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
              },
              "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                  "smithy.api#documentation": "<p>Product information.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
            "type": "structure",
            "members": {
              "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
```

```
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
            "type": "structure",
            "members": {
                "ResourceArn": {

                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
                        "smithy.api#required": {}
                    }
                },
                "SourceLicenseContext": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                    "traits": {
                        "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n      For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
                        "smithy.api#required": {}
                    }
                },
                "DestinationLicenseContext":
{
                    "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                    "traits": {
                        "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
```

the <i>License Manager User Guide</i>.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
          "type": "structure",
          "members": {
            "LicenseConversionTaskId": {
              "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

            "traits": {
                "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },

```
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a report generator.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "ReportGeneratorName":
{
            "target": "com.amazonaws.licensemanager#ReportGeneratorName",
            "traits": {
              "smithy.api#documentation": "<p>Name of the report generator.</p>",
              "smithy.api#required": {}
            }
          },
          "Type": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
              "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n    <ul>\n       <li>\n          <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n       </li>\n       <li>\n          <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n       </li>\n
</ul>",
              "smithy.api#required": {}

          }
          },
          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
              "smithy.api#required": {}
          }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
                    "smithy.api#required": {}
                  }
                },
                "ClientToken": {
                  "target": "com.amazonaws.licensemanager#ClientRequestToken",
                  "traits": {
                    "smithy.api#documentation": "<p>Unique,
case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                    "smithy.api#required": {}
                  }
                },
                "Description": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Description of the report generator.</p>"
                  }
                },
                "Tags": {
                  "target": "com.amazonaws.licensemanager#TagList",
                  "traits": {
                    "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
                  }
                }
              },
              "traits": {
                "smithy.api#input": {}
              }
            },
            "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
              "type": "structure",
              "members": {
                "LicenseManagerReportGeneratorArn":
{
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
                  }
                }
              },
              "traits": {
                "smithy.api#output": {}
              }
            },
            "com.amazonaws.licensemanager#CreateLicenseRequest": {
              "type": "structure",
```

```
        "members": {
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product
name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product SKU.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits": {
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region for the license.</p>",
              "smithy.api#required": {}

  }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
```

```
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>",
            "smithy.api#required": {}
          }
        },
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
```

```
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersion": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
```

```
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target":
"com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                        "smithy.api#required": {}
                    }
                },
                "LicenseName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License name.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ProductName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Product name.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Issuer": {
                    "target": "com.amazonaws.licensemanager#Issuer",
                    "traits":
{
                        "smithy.api#documentation": "<p>License issuer.</p>",
                        "smithy.api#required": {}
                    }
                },
                "HomeRegion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Home Region of the license.</p>",
                        "smithy.api#required": {}
                    }
                },
```

```
"Validity": {
    "target": "com.amazonaws.licensemanager#DatetimeRange",
    "traits": {
        "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
        "smithy.api#required": {}
    }
},
"LicenseMetadata": {
    "target": "com.amazonaws.licensemanager#MetadataList",
    "traits": {
        "smithy.api#documentation": "<p>Information about the license.</p>"
    }
},
"Entitlements": {
    "target": "com.amazonaws.licensemanager#EntitlementList",
    "traits": {
        "smithy.api#documentation": "<p>License entitlements.</p>",
        "smithy.api#required": {}
    }
},
"ConsumptionConfiguration": {
    "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
    "traits": {
        "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
        "smithy.api#required": {}
    }
},

"Status": {
    "target": "com.amazonaws.licensemanager#LicenseStatus",
    "traits": {
        "smithy.api#documentation": "<p>License status.</p>",
        "smithy.api#required": {}
    }
},
"ClientToken": {
    "target": "com.amazonaws.licensemanager#ClientToken",
    "traits": {
        "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
        "smithy.api#required": {}
    }
},
"SourceVersion": {
    "target": "com.amazonaws.licensemanager#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Current version of the license.</p>"
          }
        }
      }
    },
    "traits": {

"smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
    "type": "structure",
    "members": {
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>License ARN.</p>"
        }
      },
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>New version of the license.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
        "traits": {
          "smithy.api#documentation": "<p>License status.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output":
{}
    }
  },
  "com.amazonaws.licensemanager#CreateToken": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#CreateTokenRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#CreateTokenResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },

    {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateTokenRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits":
 {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
            "smithy.api#required": {}
          }
        },
        "RoleArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n        License Manager does not check whether the roles are in use.</p>"
          }
```

```
        },
        "ExpirationInDays": {
          "target": "com.amazonaws.licensemanager#Integer",
          "traits": {
            "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
            "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n        to the amr claim of the JWT token.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateTokenResponse": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",

          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
        "TokenType": {
          "target": "com.amazonaws.licensemanager#TokenType",
          "traits": {
            "smithy.api#documentation": "<p>Token type.</p>"
          }
        },
        "Token": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
          }
```

```
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DateTime": {
      "type": "timestamp"
    },
    "com.amazonaws.licensemanager#DatetimeRange": {
      "type": "structure",
      "members": {
        "Begin": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Start of the time range.</p>",
            "smithy.api#required": {}
          }
        },
        "End": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>End of the time range.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
```

```json
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
          }
        },
        "com.amazonaws.licensemanager#DeleteGrantRequest": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
              }
            },
            "StatusReason": {
              "target": "com.amazonaws.licensemanager#StatusReasonMessage",
              "traits": {
                "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Current version of the
grant.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#DeleteGrantResponse": {
```

```
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation":
"<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
        },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deletes the specified license.</p>"
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
          },

          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
          }
```

```
        },
        "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
          "type": "structure",
          "members":
{
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>ID of the license configuration.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
```

```
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
            "type": "structure",
            "members": {
                "LicenseManagerReportGeneratorArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {

    "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                        "smithy.api#required": {}
```

```
                }
              },
              "SourceVersion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Current version of the license.</p>",
                  "smithy.api#required":
{}
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#DeleteLicenseResponse": {
            "type": "structure",
            "members": {
              "Status": {
                "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
                "traits": {
                  "smithy.api#documentation": "<p>License status.</p>"
                }
              },
              "DeletionDate": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                  "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#DeleteToken": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
```

```
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RedirectException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },

        {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
            }
        },
        "com.amazonaws.licensemanager#DeleteTokenRequest": {
            "type": "structure",
            "members": {
                "TokenId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token ID.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteTokenResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}

        }
        },
        "com.amazonaws.licensemanager#DigitalSignatureMethod": {
```

```
            "type": "enum",
            "members": {
              "JWT_PS384": {
                "target": "smithy.api#Unit",
                "traits": {
                  "smithy.api#enumValue": "JWT_PS384"
                }
              }
            }
          }
        },
        "com.amazonaws.licensemanager#Entitlement": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Entitlement name.</p>",
                "smithy.api#required": {}
              }
            },
            "Value": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
              }
            },
            "MaxCount": {
              "target": "com.amazonaws.licensemanager#Long",
              "traits": {
                "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
              }
            },
            "Overage": {
              "target": "com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
              }
            },
            "Unit": {
              "target": "com.amazonaws.licensemanager#EntitlementUnit",
              "traits": {
                "smithy.api#documentation": "<p>Entitlement unit.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowCheckIn": {
              "target":
```

"com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
        }
    },
    "com.amazonaws.licensemanager#EntitlementData": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Entitlement data name.</p>",
                    "smithy.api#required": {}
                }
            },
            "Value": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Entitlement
data value.</p>"
                }
            },
            "Unit": {
                "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
                "traits": {
                    "smithy.api#documentation": "<p>Entitlement data unit.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
        }
    },
    "com.amazonaws.licensemanager#EntitlementDataList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#EntitlementData"
        }
    },
    "com.amazonaws.licensemanager#EntitlementDataUnit": {
        "type": "enum",
        "members": {

```
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {

"smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "SECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Seconds"
            }
          },
          "MICROSECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Microseconds"
            }
          },
          "MILLISECONDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Milliseconds"
            }
          },
          "BYTES": {
            "target": "smithy.api#Unit",
             "traits": {
              "smithy.api#enumValue": "Bytes"
            }
          },
          "KILOBYTES": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Kilobytes"
            }
          },
          "MEGABYTES": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Megabytes"
            }
```

```
        },
        "GIGABYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes"
          }
        },
        "TERABYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabytes"
          }
        },
        "BITS":
{
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits"
          }
        },
        "KILOBITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits"
          }
        },
        "MEGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits"
          }
        },
        "GIGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits"
          }
        },
        "TERABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits"

}
        },
        "PERCENT": {
          "target": "smithy.api#Unit",
          "traits": {
```

```
          "smithy.api#enumValue": "Percent"
      }
    },
    "BYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes/Second"
      }
    },
    "KILOBYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes/Second"
      }
    },
    "MEGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes/Second"
      }
    },
    "GIGABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
       "traits": {
        "smithy.api#enumValue": "Gigabytes/Second"
      }
    },
    "TERABYTES_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes/Second"
      }
    },
    "BITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits/Second"
      }
    },
    "KILOBITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobits/Second"
      }
    },
    "MEGABITS_PER_SECOND": {
      "target": "smithy.api#Unit",
      "traits": {
```

```
                    "smithy.api#enumValue":
"Megabits/Second"
                }
            },
            "GIGABITS_PER_SECOND": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Gigabits/Second"
                }
            },
            "TERABITS_PER_SECOND": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Terabits/Second"
                }
            },
            "COUNT_PER_SECOND": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "Count/Second"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#EntitlementList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#Entitlement"
        }
    },
    "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#EntitlementUnit": {
        "type": "enum",
        "members": {
            "COUNT": {
```

```
              "target": "smithy.api#Unit",
              "traits": {
                 "smithy.api#enumValue": "Count"
              }
           },
           "NONE": {
              "target": "smithy.api#Unit",
              "traits": {
                 "smithy.api#enumValue": "None"
              }
           },
           "SECONDS":

{
              "target": "smithy.api#Unit",
              "traits": {
                 "smithy.api#enumValue": "Seconds"
              }
           },
           "MICROSECONDS": {
              "target": "smithy.api#Unit",
              "traits": {
                 "smithy.api#enumValue": "Microseconds"
              }
           },
           "MILLISECONDS": {
              "target": "smithy.api#Unit",
              "traits": {
                 "smithy.api#enumValue": "Milliseconds"
              }
           },
           "BYTES": {
              "target": "smithy.api#Unit",
              "traits": {
                 "smithy.api#enumValue": "Bytes"
              }
           },
           "KILOBYTES": {
              "target": "smithy.api#Unit",
              "traits": {
                 "smithy.api#enumValue": "Kilobytes"
               }
           },
           "MEGABYTES": {
              "target": "smithy.api#Unit",
              "traits": {
                 "smithy.api#enumValue": "Megabytes"
              }
           },
```

```
"GIGABYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Gigabytes"
  }
},
"TERABYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Terabytes"
  }
},
"BITS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Bits"
  }
},
"KILOBITS": {
  "target": "smithy.api#Unit",
  "traits": {

    "smithy.api#enumValue": "Kilobits"
  }
},
"MEGABITS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Megabits"
  }
},
"GIGABITS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Gigabits"
  }
},
"TERABITS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Terabits"
  }
},
"PERCENT": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Percent"
  }
```

          },
          "BYTES_PER_SECOND": {
            "target":
"smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Bytes/Second"
            }
          },
          "KILOBYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Kilobytes/Second"
            }
          },
          "MEGABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Megabytes/Second"
            }
          },
          "GIGABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Gigabytes/Second"
            }
          },
          "TERABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {

"smithy.api#enumValue": "Terabytes/Second"
            }
          },
          "BITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Bits/Second"
            }
          },
          "KILOBITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Kilobits/Second"
            }
          },
          "MEGABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {

```
              "smithy.api#enumValue": "Megabits/Second"
            }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
          }
        },

    "TERABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits/Second"
          }
        },
        "COUNT_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count/Second"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsage": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage name.</p>",
            "smithy.api#required": {}
          }
        },
        "ConsumedValue": {
          "target": "com.amazonaws.licensemanager#String",

        "traits": {
            "smithy.api#documentation": "<p>Resource usage consumed.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
          }
        },
```

```
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList":
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementUsage"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
         },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
```

```
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
            }
        },
        "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
            "type": "structure",
            "members": {
                "LicenseConsumptionToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License consumption token.</p>",
                        "smithy.api#required": {}
                    }

            }
                },
                "DryRun": {
                    "target": "com.amazonaws.licensemanager#Boolean",
                    "traits": {
                        "smithy.api#default": false,
                        "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
            "type": "structure",
            "members": {
                "LicenseConsumptionToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License consumption token.</p>"
                    }
                },

                "Expiration": {
                    "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
                    }
                }
            },
            "traits": {
```

```
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#FailedDependencyException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        },
        "ErrorCode": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "FailedDependency",
          "httpResponseCode": 424
        },
        "smithy.api#documentation":
 "<p>A dependency required to run the API is missing.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 424
      }
    },
    "com.amazonaws.licensemanager#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#FilterName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
          }
        },
        "Values": {
          "target": "com.amazonaws.licensemanager#FilterValues",
          "traits": {
            "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n        describe operation. Filters can be used
to match a set of resources by specific criteria,\n        such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanager#FilterLimitExceededException": {
```

```
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "FilterLimitExceeded",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Filter"
        }
      },
      "com.amazonaws.licensemanager#FilterName": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#FilterValue": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#FilterValues": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#FilterValue",
          "traits": {
            "smithy.api#xmlName": "item"
          }
        }
      },
      "com.amazonaws.licensemanager#Filters": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Filter",
          "traits": {
            "smithy.api#xmlName": "item"
          }
        }
      },
      "com.amazonaws.licensemanager#GetAccessToken": {
        "type": "operation",
```

```
          "input": {
              "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
          },
          "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
 tokens\n        are valid for one hour.</p>"
          }
        },
      "com.amazonaws.licensemanager#GetAccessTokenRequest": {
        "type": "structure",
        "members": {
          "Token": {
            "target": "com.amazonaws.licensemanager#TokenString",
            "traits": {
              "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
              "smithy.api#required": {}
            }
          },
          "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
            "traits": {
              "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#input": { }
        }
    },
    "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
        "type": "structure",
        "members": {
            "AccessToken": {
                "target": "com.amazonaws.licensemanager#TokenString",
                "traits": {
                    "smithy.api#documentation": "<p>Temporary access token.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": { }
        }
    },
    "com.amazonaws.licensemanager#GetGrant": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetGrantRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetGrantResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"

        },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetGrantRequest": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetGrantResponse": {
          "type": "structure",
          "members": {
            "Grant": {
              "target": "com.amazonaws.licensemanager#Grant",
              "traits": {
                "smithy.api#documentation": "<p>Grant details.</p>"
              }
            }
          },
          "traits": {

"smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#GetLicense": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseRequest"
          },
          "output": {
```

```
            "target": "com.amazonaws.licensemanager#GetLicenseResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#GetLicenseConfiguration": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
    }
},
"com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
    "type": "structure",
    "members": {
        "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits":
{
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
    "type": "structure",
    "members": {
        "LicenseConfigurationId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
            }
        },
        "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
            }
        },
        "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Name of the license configuration.</p>"
            }
        },
        "Description": {
```

```
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Description of the license configuration.</p>"
        }
      },
      "LicenseCountingType": {
        "target": "com.amazonaws.licensemanager#LicenseCountingType",
        "traits": {
          "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",

        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License configuration status.</p>"
        }

      },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
        }
```

```
        },
        "ConsumedLicenseSummaryList": {
            "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
            }
        },
        "ManagedResourceSummaryList": {
            "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
                "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
            }
        },
        "ProductInformationList": {
            "target": "com.amazonaws.licensemanager#ProductInformationList",
            "traits": {
                "smithy.api#documentation": "<p>Product information.</p>"
            }
        },
        "AutomatedDiscoveryInformation": {
            "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
            "traits": {
                "smithy.api#documentation": "<p>Automated discovery information.</p>"
            }
        },
        "DisassociateWhenNotFound": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
                "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#GetLicenseConversionTask": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
    },
```

```
        "output": {
          "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {

    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
```

```
          }
        },
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
          }
        },
        "SourceLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
          }
        },
        "DestinationLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
          "traits": {
            "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
          }
        },

    "StartTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
          }
        },
        "LicenseConversionTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
          }
        },
        "EndTime": {
```

```
                        "target": "com.amazonaws.licensemanager#DateTime",
                        "traits": {
                            "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                        }
                    }
                },
                "traits": {
                    "smithy.api#output": {}
                }
            },
            "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
                "type": "operation",
                "input": {
                    "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
                },
                "output": {
                    "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
                },
                "errors": [
                    {
                        "target": "com.amazonaws.licensemanager#AccessDeniedException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#AuthorizationException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

                    },
                    {
                        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#ServerInternalException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#ValidationException"
                    }
                ],
                "traits": {
```

```
                "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
            "type": "structure",
            "members": {
                "LicenseManagerReportGeneratorArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
                        "smithy.api#required":
{}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
            "type": "structure",
            "members": {
                "ReportGenerator": {
                    "target": "com.amazonaws.licensemanager#ReportGenerator",
                    "traits": {
                        "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
```

```
                      "smithy.api#documentation": "<p>License version.</p>"
                  }
              }
          },
          "traits": {
              "smithy.api#input": {}
          }
      },
      "com.amazonaws.licensemanager#GetLicenseResponse": {
          "type": "structure",
          "members": {
              "License": {
                  "target": "com.amazonaws.licensemanager#License",
                  "traits": {
                      "smithy.api#documentation": "<p>License details.</p>"
                  }
              }
          },
          "traits": {
              "smithy.api#output": {}
          }
      },

  "com.amazonaws.licensemanager#GetLicenseUsage": {
          "type": "operation",
          "input": {
              "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
          },
          "output": {
              "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
          },
          "errors": [
              {
                  "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
                  "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
              {
                  "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                  "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                  "target": "com.amazonaws.licensemanager#ServerInternalException"
              },
              {
```

```json
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
            "type": "structure",
            "members": {
                "LicenseUsage": {
                    "target":
"com.amazonaws.licensemanager#LicenseUsage",
                    "traits": {
                        "smithy.api#documentation": "<p>License usage details.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#GetServiceSettings": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
            },
            "errors": [
                {
```

```
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "S3BucketArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"

 }
            },
            "SnsTopicArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
              }
            },
            "OrganizationConfiguration": {
              "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
              "traits": {
                "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n        cross-account discovery.</p>"
              }
            },
```

```
        "EnableCrossAccountsDiscovery": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
          }

        },
        "LicenseManagerResourceShareArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n      provides member accounts with access to this share.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Grant": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {

 "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant name.</p>",
            "smithy.api#required": {}
          }
        },
        "ParentArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Parent ARN.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>License ARN.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "GranteePrincipalArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {

"smithy.api#documentation": "<p>The grantee principal ARN.</p>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantStatus": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Grant status reason.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Granted operations.</p>",
            "smithy.api#required": {}
          }
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
```

```
              "smithy.api#documentation": "<p>The options specified for the grant.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>Describes a grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#GrantList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Grant"
        }
      },
      "com.amazonaws.licensemanager#GrantStatus": {
        "type": "enum",
        "members": {
          "PENDING_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_WORKFLOW"
            }
          },
          "PENDING_ACCEPT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_ACCEPT"
            }
          },
          "REJECTED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "REJECTED"

            }
          },
          "ACTIVE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ACTIVE"
            }
          },
          "FAILED_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED_WORKFLOW"
            }
```

```json
      },
      "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DELETED"
        }
      },
      "PENDING_DELETE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_DELETE"
        }
      },
      "DISABLED": {
        "target": "smithy.api#Unit",
        "traits":
{

          "smithy.api#enumValue": "DISABLED"
        }
      },
      "WORKFLOW_COMPLETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "WORKFLOW_COMPLETED"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#GrantedLicense": {
    "type": "structure",
    "members": {
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
        }
      },
      "LicenseName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License name.</p>"
        }

      },
      "ProductName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product name.</p>"
```

```
        }
      },
      "ProductSKU": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product SKU.</p>"
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#IssuerDetails",
        "traits": {
          "smithy.api#documentation": "<p>Granted license issuer.</p>"
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
        "traits": {
          "smithy.api#documentation": "<p>Granted license status.</p>"
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
ISO8601-UTC format.</p>"
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>"
        }
      },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
```

```
              "traits": {
                "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
              }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Granted license metadata.</p>"
            }
          },
          "CreateTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

  }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Version of the granted license.</p>"
            }
          },
          "ReceivedMetadata": {
            "target": "com.amazonaws.licensemanager#ReceivedMetadata",
            "traits": {
              "smithy.api#documentation": "<p>Granted license received metadata.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
        }
      },
      "com.amazonaws.licensemanager#GrantedLicenseList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#GrantedLicense"
        }
      },
      "com.amazonaws.licensemanager#ISO8601DateTime":
{
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
```

```
        "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-
1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
      }
    },
    "com.amazonaws.licensemanager#Integer": {
      "type": "integer"
    },
    "com.amazonaws.licensemanager#InvalidParameterValueException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InvalidParameterValueProvided",
          "httpResponseCode": 400
        },
        "smithy.api#documentation":
"<p>One or more parameter values are not valid.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#InvalidResourceStateException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InvalidResourceState",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },

    "com.amazonaws.licensemanager#InventoryFilter": {
      "type": "structure",
      "members": {
```

```
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Name of the filter.</p>",
          "smithy.api#required": {}
        }
      },
      "Condition": {
        "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
        "traits": {
          "smithy.api#documentation": "<p>Condition of the filter.</p>",
          "smithy.api#required": {}
        }
      },
      "Value": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }

    },
      "traits": {
        "smithy.api#documentation": "<p>An inventory filter.</p>"
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterCondition": {
      "type": "enum",
      "members": {
        "EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EQUALS"
          }
        },
        "NOT_EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "NOT_EQUALS"
          }
        },
        "BEGINS_WITH": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "BEGINS_WITH"
          }
        },
        "CONTAINS": {
```

```
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "CONTAINS"
                }
            }
        }
    }
},
"com.amazonaws.licensemanager#InventoryFilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#InventoryFilter"
    }
},
"com.amazonaws.licensemanager#Issuer": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Issuer name.</p>",
                "smithy.api#required": {}
            }
        },
        "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
 of sign and verify, \n      and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
    }
},
"com.amazonaws.licensemanager#IssuerDetails": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Issuer name.</p>"
            }
        },
        "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
            "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
              }
            },
            "KeyFingerprint": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
          }
        },
        "com.amazonaws.licensemanager#License": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
              }
            },
            "LicenseName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {

          "smithy.api#documentation": "<p>License name.</p>"
              }
            },
            "ProductName": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Product name.</p>"
              }
            },
            "ProductSKU": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Product SKU.</p>"
              }
            },
            "Issuer": {
              "target": "com.amazonaws.licensemanager#IssuerDetails",
              "traits": {
                "smithy.api#documentation": "<p>License issuer.</p>"
```

```
            }
        },
        "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

        "smithy.api#documentation": "<p>Home Region of the license.</p>"
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
                "smithy.api#documentation": "<p>License status.</p>"
            }
        },
        "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
                "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
            }
        },
        "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
        },
        "Entitlements": {

    "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
                "smithy.api#documentation": "<p>License entitlements.</p>"
            }
        },
        "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
            }
        },
        "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
                "smithy.api#documentation": "<p>License metadata.</p>"
            }
        },
        "CreateTime": {
```

```
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>License
creation time.</p>"
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License version.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
          }
        },
        "com.amazonaws.licensemanager#LicenseConfiguration": {
          "type": "structure",
          "members": {
            "LicenseConfigurationId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
              }
            },
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
              }
            },
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Name of the license configuration.</p>"
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Description of the license configuration.</p>"
              }
            },
            "LicenseCountingType": {
              "target": "com.amazonaws.licensemanager#LicenseCountingType",
              "traits": {
                "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
```

```
        }
      },
      "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
        }
      },
      "DisassociateWhenNotFound": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Status of the license configuration.</p>"
        }
      },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

      }
      },
      "ConsumedLicenseSummaryList": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
```

```
        "traits": {
          "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
        }
      },
      "ManagedResourceSummaryList": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
        }
      },
      "ProductInformationList": {
        "target": "com.amazonaws.licensemanager#ProductInformationList",
        "traits": {
          "smithy.api#documentation": "<p>Product information.</p>"
        }
      },
      "AutomatedDiscoveryInformation": {

        "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
        "traits": {
          "smithy.api#documentation": "<p>Automated discovery information.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n        consumed and enforced by License Manager. Components include specifications for
the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n        must be associated with a
host), and the number of licenses purchased and used.</p>"
    }
  },
  "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
    "type": "structure",
    "members": {
      "ResourceArn":
{
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Type of server resource.</p>"
        }
      },
```

```
        "ResourceOwnerId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
          }
        },
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
          }
        },
        "AmiAssociationScope": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
      "type": "enum",
      "members":
{
        "AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISABLED"
          }
        }
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits":
{

              "smithy.api#documentation": "<p>Type of resource.</p>"
            }
          },
          "ResourceStatus": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Status of the resource.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
            }
          },
          "AssociationTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
            }
          },

      "ConsumedLicenses": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurations": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfiguration"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
        "members": {
          "UsageOperation": {
            "target": "com.amazonaws.licensemanager#UsageOperation",
            "traits": {
              "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n        </p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {

    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
```

```
        },
        "SourceLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
          }
        },
        "DestinationLicenseContext": {
          "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the conversion task.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
          }
        },
        "StartTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation":
 "<p>The time the conversion task was started at.</p>"
          }
        },
        "LicenseConversionTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
          }
        },
        "EndTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
          }
        }
```

```
          },
          "traits": {
            "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
          }
        },
        "com.amazonaws.licensemanager#LicenseConversionTaskId": {
          "type": "string",
          "traits": {

          "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
          }
        },
        "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
          "type": "enum",
          "members": {
            "IN_PROGRESS": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "IN_PROGRESS"
              }
            },
            "SUCCEEDED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "SUCCEEDED"
              }
            },
            "FAILED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "FAILED"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#LicenseConversionTasks":
{
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTask"
          }
        },
        "com.amazonaws.licensemanager#LicenseCountingType": {
          "type": "enum",
```

```
        "members": {
          "VCPU": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "vCPU"
            }
          },
          "INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Instance"
            }
          },
          "CORE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Core"
            }
          },
          "SOCKET": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Socket"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseDeletionStatus": {
        "type": "enum",
        "members": {
          "PENDING_DELETE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_DELETE"
            }
          },
          "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DELETED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#License"
```

```
            }
        },
        "com.amazonaws.licensemanager#LicenseOperationFailure": {

            "type": "structure",
            "members": {
                "ResourceArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
                    }
                },
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#ResourceType",
                    "traits": {
                        "smithy.api#documentation": "<p>Resource type.</p>"
                    }
                },
                "ErrorMessage": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Error message.</p>"
                    }
                },
                "FailureTime": {
                    "target": "com.amazonaws.licensemanager#DateTime",
                    "traits": {

    "smithy.api#documentation": "<p>Failure time.</p>"
                    }
                },
                "OperationName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Name of the operation.</p>"
                    }
                },
                "ResourceOwnerId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
                    }
                },
                "OperationRequestedBy": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
```

```
            }
          },
          "MetadataList": {

              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                "smithy.api#documentation": "<p>Reserved.</p>"
              }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseOperationFailureList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecification": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

              "smithy.api#required": {}
            }
          },
          "AmiAssociationScope": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecifications": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseSpecification"
```

```
          }
        },
        "com.amazonaws.licensemanager#LicenseStatus": {
          "type": "enum",
          "members": {
            "AVAILABLE": {
              "target": "smithy.api#Unit",


              "traits": {
                  "smithy.api#enumValue": "AVAILABLE"
                }
            },
            "PENDING_AVAILABLE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "PENDING_AVAILABLE"
              }
            },
            "DEACTIVATED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "DEACTIVATED"
              }
            },
            "SUSPENDED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "SUSPENDED"
              }
            },
            "EXPIRED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "EXPIRED"
              }
            },


          "PENDING_DELETE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "PENDING_DELETE"
              }
            },
            "DELETED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "DELETED"
              }
```

```
            }
          }
        },
        "com.amazonaws.licensemanager#LicenseUsage": {
          "type": "structure",
          "members": {
            "EntitlementUsages": {
              "target": "com.amazonaws.licensemanager#EntitlementUsageList",
              "traits": {
                "smithy.api#documentation": "<p>License entitlement usages.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

          }
        },
        "com.amazonaws.licensemanager#LicenseUsageException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "LicenseUsageFailure",
              "httpResponseCode": 412
            },
            "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 412
          }
        },
        "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
          },
          "output": {

    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurationAssociations": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
          "traits": {
            "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
          }
        },
        "NextToken": {

 "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrants": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
```

```
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
            "type": "structure",
            "members": {
                "GrantArns": {
                    "target": "com.amazonaws.licensemanager#ArnList",
                    "traits": {

                    "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n                <code>LicenseArn</code>\n            </p>\n
        </li>\n          <li>\n            <p>\n                <code>GrantStatus</code>\n            </p>\n          </li>\n
  <li>\n            <p>\n                <code>GranteePrincipalARN</code>\n            </p>\n          </li>\n
<li>\n            <p>\n                <code>ProductSKU</code>\n            </p>\n          </li>\n          <li>\n
<p>\n                <code>LicenseIssuerName</code>\n
    </p>\n          </li>\n        </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
      "type": "structure",
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",

      "traits": {
              "smithy.api#documentation": "<p>Distributed grant details.</p>"
            }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
      },
      "errors": [
        {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
        "type": "structure",
        "members": {
            "LicenseOperationFailureList": {
                "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
                "traits": {
                    "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
        },

  "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configurations
for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
```

```
configurations.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#Filters",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n           <li>\n              <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n              Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n              Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n           </li>\n           <li>\n
<p>\n              <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n              Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
   </li>\n           <li>\n              <p>\n              <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n              Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n           </li>\n        </ul>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
          "type": "structure",
          "members": {
            "LicenseConfigurations": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurations",
              "traits": {
                "smithy.api#documentation": "<p>Information about the license configurations.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
```

```
        "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
      "type": "structure",
      "members": {
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
```

```
    "MaxResults": {
      "target": "com.amazonaws.licensemanager#BoxInteger",
      "traits": {

        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanager#Filters",
      "traits": {
        "smithy.api#documentation": "<p>\n        Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n      </p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
  "type": "structure",
  "members": {
    "LicenseConversionTasks": {
      "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
      "traits": {
        "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
      }
    },
    "NextToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
  },
```

```
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target":
"com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanager#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n     <ul>\n        <li>\n           <p>\n              <code>LicenseConfigurationArn</code>\n
</p>\n        </li>\n     </ul>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
```

```
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits":
{
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
      "type": "structure",
      "members": {
        "ReportGenerators": {
          "target": "com.amazonaws.licensemanager#ReportGeneratorList",
          "traits": {
            "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }

        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
```

```
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {

      "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {
```

```
      "LicenseSpecifications": {
        "target": "com.amazonaws.licensemanager#LicenseSpecifications",
        "traits": {
          "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#ListLicenseVersions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
    "type": "structure",
```

```
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#LicenseList",
          "traits": {
            "smithy.api#documentation": "<p>License details.</p>"
          }
        },

"NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenses": {
      "type": "operation",
```

```
    "input": {
       "target": "com.amazonaws.licensemanager#ListLicensesRequest"
    },
    "output": {
       "target": "com.amazonaws.licensemanager#ListLicensesResponse"
    },
    "errors": [
       {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
       },
       {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
       },
       {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
       },
       {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
       },
       {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
       },
       {
          "target": "com.amazonaws.licensemanager#ValidationException"
       }
    ],
    "traits": {
       "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
    }
 },
 "com.amazonaws.licensemanager#ListLicensesRequest": {
    "type": "structure",
    "members": {
       "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
             "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
       },

    "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
             "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n      <ul>\n         <li>\n            <p>\n               <code>Beneficiary</code>\n            </p>\n
         </li>\n         <li>\n            <p>\n               <code>ProductSKU</code>\n            </p>\n         </li>\n
   <li>\n            <p>\n               <code>Fingerprint</code>\n            </p>\n         </li>\n         <li>\n
<p>\n               <code>Status</code>\n            </p>\n         </li>\n      </ul>"
```

```
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token
for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicensesResponse": {
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#LicenseList",
            "traits": {
              "smithy.api#documentation": "<p>License details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {

  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrants": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n          recipient as this Amazon Web Services account, your organization, or an organizational unit\n
    (OU) to which this member account belongs.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
```

```
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target":
"com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
              "smithy.api#required": {}
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
    <ul>\n        <li>\n          <p>\n             <code>ParentArn</code>\n          </p>\n        </li>\n
<li>\n          <p>\n             <code>GranteePrincipalArn</code>\n          </p>\n        </li>\n      </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
```

```json
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
            "type": "structure",
            "members": {
                "Grants": {
                    "target": "com.amazonaws.licensemanager#GrantList",
                    "traits": {
                        "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
            "type": "structure",
            "members": {
                "GrantArns": {
                    "target": "com.amazonaws.licensemanager#ArnList",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n        <p>\n        <code>ProductSKU</code>\n        </p>\n
        </li>\n        <li>\n        <p>\n        <code>LicenseIssuerName</code>\n        </p>\n
</li>\n        <li>\n        <p>\n        <code>LicenseArn</code>\n        </p>\n        </li>\n
<li>\n        <p>\n        <code>GrantStatus</code>\n        </p>\n        </li>\n        <li>\n
<p>\n
        <code>GranterAccountId</code>\n        </p>\n        </li>\n        </ul>"
                    }
                },
```

```
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#MaxSize100",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
},
"com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
  "type": "structure",
  "members": {
    "Grants": {
      "target": "com.amazonaws.licensemanager#GrantList",
      "traits": {
        "smithy.api#documentation": "<p>Received grant details.</p>"
      }
    },
    "NextToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#ListReceivedLicenses": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists received licenses.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
  {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
```

```
                },
                {
```

"target": "com.amazonaws.licensemanager#ServerInternalException"

```
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
            "type": "structure",
            "members": {
                "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n           <li>\n              <p>\n                  <code>Beneficiary</code>\n              </p>\n
        </li>\n           <li>\n              <p>\n                  <code>ProductSKU</code>\n
        </p>\n           </li>\n        </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
            "type": "structure",
            "members": {
                "Licenses": {
                    "target": "com.amazonaws.licensemanager#GrantedLicenseList",
```

```
                    "traits": {
                      "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
                    }
                  },
                  "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                      "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                  }
                },
                "traits": {
                  "smithy.api#output": {}
                }
              },
              "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
                "type": "structure",
                "members": {
                  "LicenseArns": {
                    "target": "com.amazonaws.licensemanager#ArnList",
                    "traits": {
                      "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
                    }
                  },

                  "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                      "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n                <code>ProductSKU</code>\n            </p>\n
          </li>\n          <li>\n            <p>\n                <code>Status</code>\n            </p>\n          </li>\n
<li>\n            <p>\n                <code>Fingerprint</code>\n            </p>\n          </li>\n          <li>\n
<p>\n                <code>IssuerName</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Beneficiary</code>\n            </p>\n          </li>\n        </ul>"
                    }
                  },
                  "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                      "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                  },
                  "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                      "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                  }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#GrantedLicenseList",
            "traits": {
              "smithy.api#documentation": "<p>Received license details.</p>"
            }
          },

          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListResourceInventory": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FailedDependencyException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
```

```
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#InventoryFilterList",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical operators\n      are supported:</p>\n      <ul>\n         <li>\n            <p>\n               <code>account_id</code> - The ID of the Amazon Web Services account that owns the resource.\n          Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n         </li>\n         <li>\n            <p>\n               <code>application_name</code> - The name of the application. \n          Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n         </li>\n         <li>\n            <p>\n               <code>license_included</code> - The type of license included. \n          Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n          Possible values are <code>sql-server-enterprise</code> | \n          <code>sql-server-standard</code> | \n          <code>sql-server-web</code> | \n          <code>windows-server-datacenter</code>.</p>\n         </li>\n         <li>\n            <p>\n               <code>platform</code> - The platform of the resource.\n          Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n         </li>\n         <li>\n            <p>\n               <code>resource_id</code> - The ID of the resource. \n          Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n         </li>\n         <li>\n            <p>\n               <code>tag:<key></code> - The key/value combination of a tag assigned \n          to the resource. Logical operators are <code>EQUALS</code> (single account) or\n          <code>EQUALS</code> | <code>NOT_EQUALS</code> (cross account).</p>\n         </li>\n      </ul>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
        "type": "structure",
        "members": {
          "ResourceInventoryList": {
            "target": "com.amazonaws.licensemanager#ResourceInventoryList",
            "traits": {
              "smithy.api#documentation": "<p>Information
about the resources.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

        "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the tags.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListTokens": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListTokensRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListTokensResponse"
      },
```

```
        "errors":
[
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists your tokens.</p>"
    }
},
"com.amazonaws.licensemanager#ListTokensRequest": {
    "type": "structure",
    "members": {
        "TokenIds": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {

                "smithy.api#documentation": "<p>Token IDs.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n        <li>\n            <p>\n                <code>LicenseArns</code>\n            </p>\n        </li>\n
</ul>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
```

```
            "traits": {
                "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#ListTokensResponse": {
    "type": "structure",
    "members": {
        "Tokens": {
            "target": "com.amazonaws.licensemanager#TokenList",
            "traits": {
                "smithy.api#documentation": "<p>Received token details.</p>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
{
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
```

            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
             {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying license\n   consumption details by resource at a selected point in time. Use this action to audit the\n   current license consumption for any license inventory and configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
                "smithy.api#required": {}
              }
            },
             "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#Filters",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical operators\n   are supported:</p>\n   \n   \n   <p>\n<code>resourceArn</code> - The ARN of the license configuration resource.\n   Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>resourceType</code> - The resource type

(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n   Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   \n   <p>\n
   <code>resourceAccount</code> - The ID of the account that owns the resource. \n   Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   "

```
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
            "type": "structure",
            "members": {

  "LicenseConfigurationUsageList": {
                "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
                "traits": {
                  "smithy.api#documentation": "<p>Information about the license configurations.</p>"
                }
              },
              "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#Location": {
            "type": "string"
          },
          "com.amazonaws.licensemanager#Long": {
            "type": "long"
          },
          "com.amazonaws.licensemanager#ManagedResourceSummary": {
            "type": "structure",
            "members": {
              "ResourceType":
{
                "target": "com.amazonaws.licensemanager#ResourceType",
                "traits": {
                  "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
                }
```

```
      },
      "AssociationCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
    }
  },
  "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
    }
  },
  "com.amazonaws.licensemanager#MaxSize100": {

    "type": "integer",
    "traits": {
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  },
  "com.amazonaws.licensemanager#MaxSize3StringList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 3
      }
    }
  },
  "com.amazonaws.licensemanager#Message": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#Metadata": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>The key name.</p>"
          }
        },
    "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The value.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
      }
    },
    "com.amazonaws.licensemanager#MetadataList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Metadata"
      }
    },
    "com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#Options": {
      "type": "structure",
      "members": {
        "ActivationOverrideBehavior": {
          "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
          "traits": {
            "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n         Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n         <ul>\n         <li>\n            <p>As a license administrator, you can
optionally specify an\n               <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
```

\n   \n   <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n   account in your
organization.</p>\n   \n   \n   <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n   <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n   are activating the grant.</p>\n   \n   \n   <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n   <dd>\n   <p>Use this value to activate a grant
without replacing any member accounts\n   active grants for the same product.</p>\n   </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n   <dd>\n   <p>Use this value to activate a
grant and disable other active grants in any\n   member accounts
for the same product. This action will also replace their\n   previously activated grants with this activated
grant.</p>\n   </dd>\n   </dl>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n   override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager in the <i>License Manager User Guide</i>.</p>"
        }
    },
    "com.amazonaws.licensemanager#OrganizationConfiguration": {
        "type": "structure",
        "members": {
            "EnableIntegration": {
                "target": "com.amazonaws.licensemanager#Boolean",
                "traits": {
                    "smithy.api#default": false,
                    "smithy.api#documentation":
"<p>Enables Organizations integration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
        }
    },
    "com.amazonaws.licensemanager#PrincipalArnList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#Arn"
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 1
            }
        }

```
            }
        },
        "com.amazonaws.licensemanager#ProductInformation": {
            "type": "structure",
            "members": {
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Resource type. The
 possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ProductInformationFilterList": {
                    "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n        <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n        <p>The following filters and are supported when
the resource type \n        is <code>SSM_MANAGED</code>:</p>\n
        <ul>\n        <li>\n          <p>\n            <code>Application Name</code> - The name of the
application.\n          Logical operator is <code>EQUALS</code>.</p>\n          </li>\n        <li>\n
<p>\n            <code>Application Publisher</code> - The publisher of the application.\n          Logical
operator is <code>EQUALS</code>.</p>\n          </li>\n        <li>\n          <p>\n
<code>Application Version</code> - The version of the application.\n          Logical operator is
<code>EQUALS</code>.</p>\n          </li>\n        <li>\n          <p>\n            <code>Platform
Name</code> - The name of the platform.\n          Logical operator is <code>EQUALS</code>.</p>\n
</li>\n        <li>\n          <p>\n            <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
        </li>\n        <li>\n          <p>\n            <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>.  The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n          </p>\n          </li>\n        <li>\n          <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n          Logical operator is <code>NOT_EQUALS</code>.</p>\n          </li>\n
<li>\n          <p>\n            <code>License Included</code> - The type of license included.\n          Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n          Possible values are:
 <code>sql-server-enterprise</code> | \n          <code>sql-server-standard</code> | \n          <code>sql-server-
web</code> | \n          <code>windows-server-datacenter</code>.</p>\n          </li>\n        </ul>\n
<p>The following filters and logical operators are supported when the resource type\n        is
<code>RDS</code>:</p>\n        <ul>\n        <li>\n          <p>\n            <code>Engine Edition</code> -
The edition of the database engine.\n          Logical operator is <code>EQUALS</code>.\n          Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n          </li>\n        <li>\n          <p>\n            <code>License Pack</code> - The license
```

pack.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>data guard</code> | \n   <code>diagnostic pack sqlt</code> | \n   <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n   <code>olap</code>.</p>\n
\n   ",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#ProductInformationFilter": {
        "type": "structure",
        "members": {
            "ProductInformationFilterName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Filter name.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductInformationFilterValue": {
                "target": "com.amazonaws.licensemanager#StringList",

                "traits": {
                    "smithy.api#documentation": "<p>Filter value.</p>"
                }
            },
            "ProductInformationFilterComparator": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Logical operator.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes product information filters.</p>"
        }
    },
    "com.amazonaws.licensemanager#ProductInformationFilterList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#ProductInformationFilter"
        }
    },
    "com.amazonaws.licensemanager#ProductInformationList": {

```
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#ProductInformation"
          }
        },
        "com.amazonaws.licensemanager#ProvisionalConfiguration": {
          "type": "structure",
          "members": {
            "MaxTimeToLiveInMinutes": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#RateLimitExceededException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "RateLimitExceeded",
              "httpResponseCode": 429
            },
            "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 429
          }
        },
        "com.amazonaws.licensemanager#ReceivedMetadata": {
          "type": "structure",
          "members": {
            "ReceivedStatus": {
              "target": "com.amazonaws.licensemanager#ReceivedStatus",
              "traits": {
                "smithy.api#documentation": "<p>Received status.</p>"
              }
            },
            "ReceivedStatusReason": {
```

```
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>Received
status reason.</p>"
            }
          },
          "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed operations.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
        }
      },
      "com.amazonaws.licensemanager#ReceivedStatus": {
        "type": "enum",
        "members": {
          "PENDING_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_WORKFLOW"
            }
          },
          "PENDING_ACCEPT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue":
"PENDING_ACCEPT"
            }
          },
          "REJECTED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "REJECTED"
            }
          },
          "ACTIVE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ACTIVE"
            }
          },
          "FAILED_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
```

```
                "smithy.api#enumValue": "FAILED_WORKFLOW"
              }
            },
            "DELETED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "DELETED"
              }
            },
            "DISABLED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "DISABLED"
              }
            },
            "WORKFLOW_COMPLETED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "WORKFLOW_COMPLETED"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#RedirectException": {
          "type": "structure",
          "members": {
            "Location": {
              "target": "com.amazonaws.licensemanager#Location",
              "traits": {
                "smithy.api#httpHeader": "Location"
              }
            },
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 308
          }
        },
        "com.amazonaws.licensemanager#RejectGrant": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#RejectGrantRequest"
          },
          "output": {
```

```
              "target": "com.amazonaws.licensemanager#RejectGrantResponse"
            },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
          }
        },
        "com.amazonaws.licensemanager#RejectGrantRequest": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        }
      },
      "com.amazonaws.licensemanager#RejectGrantResponse": {
        "type": "structure",
```

```
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant ARN.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant version.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#RenewType":
{
            "type": "enum",
            "members": {
                "NONE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "None"
                    }
                },
                "WEEKLY": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Weekly"
                    }
                },
                "MONTHLY": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Monthly"
                    }
                }
            }
        },
```

```
"com.amazonaws.licensemanager#ReportContext": {
    "type": "structure",
    "members": {
        "licenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {

                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
    }
},
"com.amazonaws.licensemanager#ReportFrequency": {
    "type": "structure",
    "members": {
        "value": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
                "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
            }
        },
        "period": {
            "target": "com.amazonaws.licensemanager#ReportFrequencyType",

            "traits": {
                "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
    }
},
"com.amazonaws.licensemanager#ReportFrequencyType": {
    "type": "enum",
    "members": {
        "DAY": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "DAY"
            }
```

```
        },
        "WEEK": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WEEK"
          }
        },
        "MONTH": {
          "target": "smithy.api#Unit",
          "traits":
{

            "smithy.api#enumValue": "MONTH"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportGenerator": {
      "type": "structure",
      "members": {
        "ReportGeneratorName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>"
          }
        },
        "ReportType": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
          }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
          }
        },
        "ReportFrequency": {
          "target": "com.amazonaws.licensemanager#ReportFrequency",
          "traits": {
            "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
          }
        },
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
```

```
          }
        },
        "LastRunStatus": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
          }
        },
        "LastRunFailureReason":
{

          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
          }
        },
        "LastReportGenerationTime": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
          }
        },
        "ReportCreatorAccount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",

      "traits": {
            "smithy.api#documentation": "<p>Description of the report generator.</p>"
          }
        },
        "S3Location": {
          "target": "com.amazonaws.licensemanager#S3Location",
          "traits": {
            "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
          }
        },
        "CreateTime": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the report was created.</p>"
          }
        },
```

```
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
    }
  },
  "com.amazonaws.licensemanager#ReportGeneratorList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ReportGenerator"
    }
  },
  "com.amazonaws.licensemanager#ReportGeneratorName": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "com.amazonaws.licensemanager#ReportType": {
    "type": "enum",
    "members": {
      "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
        }
      },
      "LICENSE_CONFIGURATION_USAGE_REPORT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "LicenseConfigurationUsageReport"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#ReportTypeList": {
    "type": "list",
    "member": {
```

```
          "target": "com.amazonaws.licensemanager#ReportType"
        }
      },
      "com.amazonaws.licensemanager#ResourceInventory": {
        "type": "structure",
        "members": {
          "ResourceId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits":
{

              "smithy.api#documentation": "<p>Type of resource.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "Platform": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Platform of the resource.</p>"
            }
          },
          "PlatformVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
            }
          },
          "ResourceOwningAccountId":
{

            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about a resource.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanager#ResourceInventoryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#ResourceInventory"
        }
    },
    "com.amazonaws.licensemanager#ResourceLimitExceededException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "ResourceLimitExceeded",

                "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "InvalidResource.NotFound",
                "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>The resource cannot be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#ResourceType": {
        "type": "enum",
        "members": {
            "EC2_INSTANCE": {
                "target": "smithy.api#Unit",
                "traits": {
```

```
          "smithy.api#enumValue": "EC2_INSTANCE"
        }
      },
      "EC2_HOST": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "EC2_HOST"
        }
      },
      "EC2_AMI": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "EC2_AMI"
        }
      },
      "RDS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "RDS"
        }
      },
      "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
        }
      }
    }
  }
},
"com.amazonaws.licensemanager#S3Location": {
  "type": "structure",
  "members": {
    "bucket": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
      }
    },
    "keyPrefix": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#ServerInternalException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InternalError",
            "httpResponseCode": 500
          },
          "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanager#SignedToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 4096
          }
        }
      },
      "com.amazonaws.licensemanager#StatusReasonMessage": {
        "type": "string",
        "traits":
{
          "smithy.api#length": {
            "min": 0,
            "max": 400
          },
          "smithy.api#pattern": "^[\\s\\S]+$"
        }
      },
      "com.amazonaws.licensemanager#String": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#StringList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "com.amazonaws.licensemanager#Tag": {
```

```json
      "type": "structure",
      "members": {
        "Key": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Tag key.</p>"
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation":
"<p>Tag value.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#TagKeyList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#String"
      }
    },
    "com.amazonaws.licensemanager#TagList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Tag"
      }
    },
    "com.amazonaws.licensemanager#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#TagResourceResponse"
      },
      "errors": [
        {
          "target":
"com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
```

```
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits":
 {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>One or more tags.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#TokenData": {
        "type": "structure",
        "members": {
```

```
      "TokenId": {
        "target": "com.amazonaws.licensemanager#String",
         "traits": {
           "smithy.api#documentation": "<p>Token ID.</p>"
        }
      },
      "TokenType": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
           "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
        }
      },
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
           "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
        }
      },
      "ExpirationTime": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
           "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
 format.</p>"
        }
      },
      "TokenProperties": {
        "target": "com.amazonaws.licensemanager#MaxSize3StringList",
        "traits": {
           "smithy.api#documentation": "<p>Data specified by the caller.</p>"
        }
      },
      "RoleArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
           "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
           "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
        }
      }
    },
    "traits": {
```

```
            "smithy.api#documentation": "<p>Describes a token.</p>"
          }
        },
        "com.amazonaws.licensemanager#TokenList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#TokenData"
          }
        },
        "com.amazonaws.licensemanager#TokenString": {
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 4096
            },
            "smithy.api#pattern": "^\\S+$"
          }
        },
        "com.amazonaws.licensemanager#TokenType": {
          "type": "enum",
          "members": {
            "REFRESH_TOKEN": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "REFRESH_TOKEN"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
  {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#UntagResource": {
          "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanager#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanager#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
```

```
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
        },
        "output": {

            "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
```

```
        "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
        }
    },
    "LicenseConfigurationStatus": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
        "traits": {
            "smithy.api#documentation": "<p>New status of the license configuration.</p>"
        }
    },
    "LicenseRules": {
        "target": "com.amazonaws.licensemanager#StringList",
        "traits": {

    "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
    configuration is licenseAffinityToHost.</p>"
        }
    },
    "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
            "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
        }
    },
    "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
            "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
        }
    },
    "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
            "smithy.api#documentation": "<p>New name
 of the license configuration.</p>"
        }
    },
    "Description": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
            "smithy.api#documentation": "<p>New description of the license configuration.</p>"
        }
    },
    "ProductInformationList": {
        "target": "com.amazonaws.licensemanager#ProductInformationList",
```

```
                    "traits": {
                        "smithy.api#documentation": "<p>New product information.</p>"
                    }
                },
                "DisassociateWhenNotFound": {
                    "target": "com.amazonaws.licensemanager#BoxBoolean",
                    "traits": {
                        "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                    }
                }
            },
            "traits":
{
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
```

```
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
            "smithy.api#required": {}
          }
        },
        "ReportGeneratorName": {
          "target": "com.amazonaws.licensemanager#ReportGeneratorName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>",
            "smithy.api#required": {}
          }
        },
        "Type": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n        <ul>\n          <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n          </li>\n          <li>\n            <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n          </li>\n        </ul>",
            "smithy.api#required": {}
          }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The report context.</p>",
                "smithy.api#required": {}
              }
            },
            "ReportFrequency": {
              "target": "com.amazonaws.licensemanager#ReportFrequency",
              "traits":

{

                "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                "smithy.api#required": {}
              }
            },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientRequestToken",
              "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":

{

        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
        },
        "errors": [
```

```
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },

          {
            "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
          },
          {
            "target": "com.amazonaws.licensemanager#LicenseUsageException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
        }
      },

    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
              "smithy.api#required": {}
            }
          },
          "AddLicenseSpecifications": {
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
              "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
            }
          },
```

```json
        "RemoveLicenseSpecifications": {
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
                "smithy.api#documentation":
"<p>ARNs of the license configurations to remove.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
    }
},
```

```
    "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
          }
        },
        "SnsTopicArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
          }
        },
        "OrganizationConfiguration": {
          "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
          }
        },
        "EnableCrossAccountsDiscovery": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

        }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UsageOperation": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
```

```
                  "min": 0,
                  "max": 50
              }
          }
      },
      "com.amazonaws.licensemanager#ValidationException": {
          "type": "structure",
          "members": {
              "Message": {
                  "target": "com.amazonaws.licensemanager#Message"
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

          "smithy.api#error": "client",
              "smithy.api#httpError": 400
          }
      }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
 params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
 return nil, err
 }
```

```go
    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version
    of the license.
    //
    // This member is required.
    SourceVersion *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
```

```go
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack,
options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
```

```
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
 }

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
  }
 }
```

1.33 aws-aws-sdk-go-v2-internal-endpoints 2.5.7

1.33.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
 )
```

```go
// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil,
 err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```go
  noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err !=
 nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack);
 err != nil {
  return err
```

```
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack,
options); err != nil {
return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:       region,
ServiceID:     ServiceID,
OperationName: "AssociateLicense",
}
}
```

AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
name: License Scan

on: [pull_request]

jobs:
 licensescan:
  name: License Scan
  runs-on: ubuntu-latest
  strategy:
   matrix:
    python-version: [3.9]

  steps:
   - name: Checkout target
     uses: actions/checkout@v2
     with:
      path: sdkbase
      ref: ${{ github.base_ref }}
   - name: Checkout this ref
     uses: actions/checkout@v2
     with:
      path: new-ref
      fetch-depth: 0
   - name: Get Diff
     run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }} ${{ github.sha }} > refDiffFiles.txt
   - name: Get Target Files
     run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
   - name: Checkout scancode
     uses: actions/checkout@v2
     with:
      repository: nexB/scancode-toolkit
      path: scancode-toolkit
      fetch-depth: 1
   - name: Set up Python ${{ matrix.python-version }}
     uses: actions/setup-python@v2
     with:
      python-version: ${{ matrix.python-version }}
   # ScanCode
   - name: Self-configure scancode
     working-directory: ./scancode-toolkit
     run: ./scancode --help
   - name: Run Scan code on target
     run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt

---

```yaml
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary
```

```go
	*string

	noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
```

```go
	return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
	return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
	return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
	return err
	}
	if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
	return
	err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
	return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
	return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
	return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
	return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
	return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
```

```
return &awsmiddleware.RegisterServiceMetadata{
    Region:        region,
    ServiceID:      ServiceID,
    OperationName: "CheckInLicense",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
            "type": "structure",
            "members": {
                "Name": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
                    }
                },
                "Values": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                    "traits": {
                        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
                    }
                },
                "Operator": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
                    "traits": {
                        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
            }
        },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
    "type": "structure",
    "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
```

settings include if\n   Organizations integration has been enabled, and
which Regions data will be aggregated from.</p>"
                }
              },
              "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
              },
              "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                  "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
              },
              "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                  "smithy.api#documentation": "<p>The Region in which License Manager displays
the aggregated data for Linux\n   subscriptions.</p>"
                }
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
          "type": "structure",
          "members": {
            "AmiId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
              }
            },
            "InstanceID": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
              }
            },
            "InstanceType": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
              }

```
        },
        "AccountID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the instance.</p>"
          }
        },
        "Region": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
          }
        },
        "UsageOperation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
          }
        },
        "ProductCode": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
          "traits": {
            "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
          }
        },
        "LastUpdatedTime": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
          }
        },
        "SubscriptionName": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
          }
```

```
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
      ],

      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager Linux Subscriptions",
          "arnNamespace": "license-manager-linux-subscriptions",
          "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
```

                },
          "aws.auth#sigv4": {
                "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
                "additionalAllowedHeaders": [
                      "*",
                      "content-type",
                      "x-amz-content-sha256",
                      "x-amz-user-agent",
                      "x-amzn-platform-id",
                      "x-amzn-trace-id",
                      "content-length",
                      "x-api-key",
                      "authorization",
                      "x-amz-date",

   "x-amz-security-token",
                      "Access-Control-Allow-Headers",
                      "Access-Control-Allow-Methods",
                      "Access-Control-Allow-Origin"
                ],
                "additionalExposedHeaders": [
                      "x-amzn-errortype",
                      "x-amzn-requestid",
                      "x-amzn-trace-id"
                ]
          },
          "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n   Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
                "version": "1.0",
                "parameters": {
                      "Region": {
                            "builtIn": "AWS::Region",
                            "required": false,
                            "documentation": "The
 AWS region used to dispatch the request.",
                            "type": "String"
                      },
                      "UseDualStack": {
                            "builtIn": "AWS::UseDualStack",
                            "required": true,
                            "default": false,
                            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",

```
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

                "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }
            },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {

                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
                        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                        "type": "error"
                        },
```

```
            {
                "conditions": [],
                "type": "tree",
                "rules": [
                    {

                "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],

            "endpoint": {
                        "url": {
                            "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        }
    ]
},
{
    "conditions": [],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",

"argv": [
                    {
                        "ref": "Region"
```

```
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "aws.partition",
                      "argv": [
                        {
                          "ref": "Region"
                        }
                      ],
"assign": "PartitionResult"
                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseFIPS"
                            },
                            true
                          ]
                        },
                {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }
                      ],
                      "type": "tree",
                      "rules": [
                        {
                          "conditions": [
                            {
```

```
                        "fn": "booleanEquals",
                            "argv": [
                              true,
                              {
                                 "fn": "getAttr",
                                 "argv": [
                                    {
                                       "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                 ]

        }

                              ]
                           },
                           {
                              "fn": "booleanEquals",
                              "argv": [
                                 true,
                                 {
                                    "fn": "getAttr",
                                    "argv": [
                                       {
                                          "ref": "PartitionResult"
                                       },

                              "supportsDualStack"
                                 ]
                              }
                           ]
                        }
                     ],
                     "type": "tree",
                     "rules": [
                        {
                           "conditions": [],
                           "type": "tree",
                           "rules": [
                              {

                  "conditions": [],
                                 "endpoint": {
                                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                 },
```

```
                                    "type": "endpoint"
                                }
                            ]
                        }
                    ]

                },
                    {
                        "conditions": [],
                        "error": "FIPS and DualStack are enabled, but this partition does not support

one or both",

                        "type": "error"
                    }
                ]
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {

                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,

        {

                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsFIPS"
                                        ]
                                    }
                                ]
                            }
```

```
                                          ],
                                          "type": "tree",

                                "rules": [
                                        {
                                           "conditions": [],
                                           "type": "tree",
                                           "rules": [
                                             {
                                                "conditions": [],
                                                "endpoint": {
                                                   "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                                                    "properties": {},
                                                    "headers": {}

                                                 },
                                                 "type": "endpoint"
                                             }
                                          ]
                                        }
                                   ]
                              },
                              {
                                 "conditions": [],
                                 "error": "FIPS is enabled but this partition does not support FIPS",
                                 "type": "error"
                              }
                           ]

          },
                              {
                                 "conditions": [
                                    {
                                       "fn": "booleanEquals",
                                       "argv": [
                                          {
                                             "ref": "UseDualStack"
                                          },
                                          true
                                       ]
                                    }
                                 ],
                                 "type": "tree",
                                 "rules": [
                                    {
```

```
"conditions": [
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
],
"type": "tree",
"rules": [
    {
        "conditions": [],
        "type": "tree",
        "rules": [
            {
                "conditions": [],

                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ]
    }
]
},

{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
]
},
{
    "conditions": [],
```

```
                              "type": "tree",
                              "rules": [
                                 {
                                    "conditions": [],
                                    "endpoint": {
                                       "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                       "properties": {},
                                       "headers": {}
                                    },
                                    "type": "endpoint"
                                 }
                              ]
                           }
                        ]
                     }
                  ]
               },
               {
                  "conditions": [],
                  "error": "Invalid Configuration: Missing Region",
                  "type": "error"
               }
            ]
         }
      ]
   },
   "smithy.rules#endpointTests": {
      "testCases": [
         {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
               "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
               }
            },
            "params": {
               "Region": "us-east-1",
               "UseFIPS": true,
               "UseDualStack": true
            }
         },
         {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {

   "endpoint": {
               "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
```

```
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {

      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
```

```
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint":
{

            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
```

```
      "Region": "us-gov-east-1",
      "UseFIPS": true,
       "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint":
{

        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
       "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
```

```
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                    }
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled but this partition does not support DualStack"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {

"documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
                    }
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
```

```
      "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
      },
      "params": {

    "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
      "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",

    "UseFIPS": false,
        "UseDualStack": false
      }
```

```
      },
      {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
          "endpoint": {
            "url": "https://example.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false,
          "Endpoint": "https://example.com"
        }
      },
      {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",

        "expect": {
          "endpoint": {
            "url": "https://example.com"
          }
        },
        "params": {
          "UseFIPS": false,
          "UseDualStack": false,
          "Endpoint": "https://example.com"
        }
      },
      {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {
          "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false,

      "Endpoint": "https://example.com"
        }
      },
      {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
```

```
                    "params": {
                       "Region": "us-east-1",
                       "UseFIPS": false,
                       "UseDualStack": true,
                       "Endpoint": "https://example.com"
                    }
                 },
                 {
                    "documentation": "Missing region",
                    "expect": {
                       "error": "Invalid Configuration: Missing Region"
                    }
                 }

          }
             ],
             "version": "1.0"
          }
       }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
       "type": "string",
       "traits": {
          "smithy.api#enum": [
             {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
             },
             {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
             }
          ]
       }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
       "type": "structure",
       "members": {
          "SourceRegions": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
             }
          },
          "OrganizationIntegration": {
```

```
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
          "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
      "type":
"operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with
commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {

    "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Instances"
        }
```

```
                }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
                "type": "structure",
                "members": {
                    "Filters": {
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                        "traits": {
                            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n        more sets of key-value pairs that you specify. For example, you can filter by
 the name of\n        <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n        <p>The valid names for this filter
are:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n
</li>\n            <li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n            </li>\n
<li>\n                <p>\n                    <code>AccountID</code>\n                </p>\n            </li>\n            <li>\n
<p>\n                    <code>Status</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Region</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>UsageOperation</code>\n                </p>\n            </li>\n            <li>\n
  <p>\n                    <code>ProductCode</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
   <code>InstanceType</code>\n                </p>\n            </li>\n        </ul>\n        <p>The valid Operators for this
filter are:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>contains</code>\n                </p>\n
</li>\n            <li>\n                <p>\n                    <code>equals</code>\n                </p>\n            </li>\n            <li>\n
    <p>\n                    <code>Notequal</code>\n                </p>\n            </li>\n        </ul>"
                        }
                    },
                    "MaxResults": {
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                        "traits": {
                            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

                        }
                    },
                    "NextToken": {
                        "target": "smithy.api#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                            "smithy.api#length": {
                                "min": 1,
                                "max": 16384
                            }
                        }
                    }
                },
                "traits": {
                    "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
                }
            },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n      organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
```

```
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
        }

    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n     partially match, or don't
match a certain subscription's name.</p>\n        <p>The valid names for this filter are:</p>\n        <ul>\n
<li>\n            <p>\n                <code>Subscription</code>\n                </p>\n            </li>\n        </ul>\n
<p>The valid Operators for this filter are:</p>\n
        <ul>\n            <li>\n                <p>\n                <code>contains</code>\n                </p>\n            </li>\n
<li>\n            <p>\n                <code>equals</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
        <code>Notequal</code>\n                </p>\n            </li>\n        </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                "smithy.api#length": {

                    "min": 1,
                    "max": 16384
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
    }
},
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},

"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
```

```
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [

{
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits":
{
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"

            }
        },
        "Type": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n     Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n     Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
            }
        },
```

```
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },

  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/subscription/UpdateServiceSettings"
                },
                "smithy.api#idempotent": {}
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
            "type": "structure",
            "members": {
                "LinuxSubscriptionsDiscovery": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                    "traits": {
                        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                        "smithy.api#required": {}
                    }
                },
                "LinuxSubscriptionsDiscoverySettings": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                    "traits": {
                        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",

        "smithy.api#required": {}
                    }
                },
                "AllowUpdate": {
                    "target": "smithy.api#Boolean",
                    "traits": {
                        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
            "type": "structure",
            "members": {
                "LinuxSubscriptionsDiscovery": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },

      "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
          "smithy.api#error": "client"
        }
      }
```

```go
        }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //   - Beneficiary
 //   - ProductSKU
 //   - Fingerprint
 //
   - Status
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
```

```go
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListLicensesOutput struct {

	// License details.
	Licenses []types.License

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	if
	err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
```

```go
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
		err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
		err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
```

```go
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
    ServiceID,
  OperationName: "ListLicenses",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
```

//
//
This member is required.
LicenseType types.LicenseType

// The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
 }
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
 }
if err := addProtocolFinalizerMiddlewares(stack,
options, "DisassociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
 }

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
 }
if err = addSetLoggerMiddleware(stack, options); err != nil {

```go
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
        return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
        return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
    err != nil {
        return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
        return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
        return err
    }
    if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
        return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
        return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
        return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
        return err
```

```go
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
 {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
```

```go
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type
 CheckoutLicenseInput struct {

 // Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {
```

```go
	// Checkout type.
	CheckoutType types.CheckoutType

	// Allowed license entitlements.
	EntitlementsAllowed []types.EntitlementData

	// Date and time at which the license checkout expires.
	Expiration *string

	// Date and time at which the license checkout is issued.
	IssuedAt *string

	// Amazon Resource
	Name (ARN) of the checkout license.
	LicenseArn *string

	// License consumption token.
	LicenseConsumptionToken *string

	// Node ID.
	NodeId *string

	// Signed token.
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil
```

```
{
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack);
err != nil {
return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
```

```go
    return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
    return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
    return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
  }
  return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata {
  return &awsmiddleware.RegisterServiceMetadata{
    Region:        region,
    ServiceID:     ServiceID,
    OperationName: "CheckoutLicense",
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
  "context"
  "fmt"
  awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
  "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
  "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
  "github.com/aws/smithy-go/middleware"
  smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
  if params == nil {
    params = &CheckoutBorrowLicenseInput{}
  }

  result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
```

```go
	if err != nil {
		return nil, err
	}

	out := result.(*CheckoutBorrowLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type CheckoutBorrowLicenseInput struct {

	// Unique, case-sensitive identifier
	that you provide to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// Digital signature method. The possible value is JSON Web Signature (JWS)
	// algorithm PS384. For more information, see RFC 7518 Digital Signature with
	// RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
	//
	// This member is required.
	DigitalSignatureMethod types.DigitalSignatureMethod

	// License entitlements. Partial checkouts are not supported.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Amazon Resource Name (ARN) of the license. The license must use the borrow
	// consumption configuration.
	//
	// This member is required.
	LicenseArn *string

	// Information about constraints.
	CheckoutMetadata []types.Metadata

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

	// Information about constraints.
```

CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
 }
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
if err = addClientUserAgent(stack, options); err != nil {
 return
err
 }
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
 }
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
```

```go
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
    }

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
    Region:        region,
    ServiceID:     ServiceID,
    OperationName: "CheckoutBorrowLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
    params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
    c.addOperationCreateLicenseMiddlewares)
```

```
if err != nil {
 return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

// License beneficiary.
//
// This member is required.
Beneficiary *string

// Unique, case-sensitive identifier that you provide
to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
```

```go
	LicenseName *string

	// Product name.
	//
	// This member is required.
	ProductName *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// Date and time range during which the license is valid, in ISO8601-UTC format.
	//
	// This member is required.
	Validity
	*types.DatetimeRange

	// Information about the license.
	LicenseMetadata []types.Metadata

	noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
	return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
	return err
```

```go
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
		return fmt.Errorf("add
protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
```

```go
   return err
  }
  if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
   return err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
   return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
   return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
   return err
  }
  if err = addResponseErrorMiddleware(stack); err != nil {
   return err
  }
  if err = addRequestResponseLogging(stack, options); err != nil {
   return err
  }
  if err = addDisableHTTPSMiddleware(stack, options); err != nil {
   return err
  }
  return nil
 }

 func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
  return &awsmiddleware.RegisterServiceMetadata{
   Region:       region,
   ServiceID:    ServiceID,
   OperationName:
  "CreateLicense",
  }
 }
 // Code generated by smithy-go-codegen DO NOT EDIT.

 package licensemanager

 import (
  "context"
  "fmt"
  awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
  "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
  "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
  "github.com/aws/smithy-go/middleware"
  smithyhttp "github.com/aws/smithy-go/transport/http"
 )
```

```go
// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
	if params == nil {
		params = &GetLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*GetLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type GetLicenseInput struct {

	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// License version.
	Version
	*string

	noSmithyDocumentSerde
}

type GetLicenseOutput struct {

	// License details.
	License *types.License

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
```

```go
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return
err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```go
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpGetLicenseValidationMiddleware(stack);
    err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
    err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
    return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
    return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
    Region:        region,
    ServiceID:     ServiceID,
    OperationName: "GetLicense",
    }
}
```

distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```go
type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
      - ProductSKU
    //   - Status
    //   - Fingerprint
    //   - IssuerName
    //   - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
```

```
	middleware.After)
	if err != nil {
	return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
	return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
	return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack);
	err != nil {
	return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
	return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
	return err
	}
```

```
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:     ServiceID,
 OperationName: "ListReceivedLicenses",
 }
}
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding

those notices that do not

      pertain to any part of the Derivative Works, in at least one
      of the following places: within a NOTICE text file distributed
      as part of the Derivative Works; within the Source form or
      documentation, if provided along with the Derivative Works; or,
      within a display generated by the Derivative Works, if and
      wherever such third-party notices normally appear. The contents
      of the NOTICE file are for informational purposes only and
      do not modify the License. You may add Your own attribution
      notices within Derivative Works that You distribute, alongside
      or as an addendum to the NOTICE text from the Work, provided
      that such additional attribution notices cannot be construed
      as modifying the License.

  You may add Your own copyright statement to Your modifications and
  may provide additional or different license terms and conditions

 for use, reproduction, or distribution of Your modifications, or
  for any such Derivative Works as a whole, provided Your use,
  reproduction, and distribution of the Work otherwise complies with
  the conditions stated in this License.

 5. Submission of Contributions. Unless You explicitly state otherwise,
  any Contribution intentionally submitted for inclusion in the Work
  by You to the Licensor shall be under the terms and conditions of
  this License, without any additional terms or conditions.
  Notwithstanding the above, nothing herein shall supersede or modify
  the terms of any separate license agreement you may have executed
  with Licensor regarding such Contributions.

 6. Trademarks. This License does not grant permission to use the trade
  names, trademarks, service marks, or product names of the Licensor,
  except as required for reasonable and customary use in describing the
  origin of the Work and reproducing the
content of the NOTICE file.

 7. Disclaimer of Warranty. Unless required by applicable law or
  agreed to in writing, Licensor provides the Work (and each
  Contributor provides its Contributions) on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
  implied, including, without limitation, any warranties or conditions
  of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
  PARTICULAR PURPOSE. You are solely responsible for determining the
  appropriateness of using or redistributing the Work and assume any
  risks associated with Your exercise of permissions under this License.

 8. Limitation of Liability. In no event and under no legal theory,
  whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
 incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```json
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
                "smithy.api#error": "client"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
            "type": "structure",
            "members": {
                "DirectoryId": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation":
"<p>Details about an Active Directory identity provider.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
```

```
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n      <note>\n         <p>Your estimated bill for charges on the number of users and related
costs will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
    monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n      </note>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n    resource.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
```

```
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/DisassociateUser"
        },
        "smithy.api#idempotent": {}
      }

    },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },


  "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used
to return more specific results from a describe\n     operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n     attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
      }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
            "ActiveDirectoryIdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n     settings such as the subnets to provision VPC endpoints.</p>",
                    "smithy.api#required":
{}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Describes an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status
 of an EC2 instance resource.</p>",
            "smithy.api#required": {}
          }
        },
        "Products": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
            "smithy.api#required": {}
          }
        },
        "LastStatusCheckDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
```

```
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                }
            }

        }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits":
{
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
```

```
        "traits": {
          "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
          "smithy.api#required": {}
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the user.</p>"
        }
      },
      "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
      },
      "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        }
      }
    },
      "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",

    "version": "2018-05-10",
        "operations": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
          },

},
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
```

```
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
      },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
          "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id",
              "content-length",
              "x-api-key",
              "authorization",
              "x-amz-date",
              "x-amz-security-token",
              "Access-Control-Allow-Headers",
              "Access-Control-Allow-Methods",
              "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"

  ]
      },
      "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
      "smithy.api#title": "AWS License Manager User Subscriptions",
      "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
              "Region": {
                  "builtIn": "AWS::Region",
                  "required": false,
                  "documentation": "The AWS region used to dispatch the request.",
                  "type": "String"
              },
              "UseDualStack": {
                  "builtIn": "AWS::UseDualStack",
                  "required": true,
                  "default": false,
```

```
        "documentation": "When true, use the dual-stack
endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }

    },
        "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },

                                true
                            ]
                        }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
```

```
                                    "type": "error"
                          },
                          {
                              "conditions": [
                                  {
                                      "fn": "booleanEquals",
                                      "argv": [
                                          {
                                              "ref": "UseDualStack"
                                          },
                                          true
                                      ]
                                  }
                              ],

"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                              "type": "error"
                          },
                          {
                              "conditions": [],
                              "endpoint": {
                                  "url": {
                                      "ref": "Endpoint"
                                  },
                                  "properties": {},
                                  "headers": {}
                              },
                              "type": "endpoint"
                          }
                      ]
                  },
                  {
                      "conditions": [
                          {
                              "fn": "isSet",
                              "argv": [

{
                                      "ref": "Region"
                                  }
                              ]
                          }
                      ],
                      "type": "tree",
                      "rules": [
                          {
                              "conditions": [
                                  {
```

```
                              "fn": "aws.partition",
                              "argv": [
                                {
                                   "ref": "Region"
                                }
                              ],
                              "assign": "PartitionResult"
                           }
                        ],
                        "type": "tree",

              "rules": [
                           {
                              "conditions": [
                                 {
                                    "fn": "booleanEquals",
                                    "argv": [
                                       {
                                          "ref": "UseFIPS"
                                       },
                                       true
                                    ]
                                 },
                                 {
                                    "fn": "booleanEquals",
                                    "argv": [
                                       {
                                          "ref": "UseDualStack"

                                       },
                                       true
                                    ]
                                 }
                              ],
                              "type": "tree",
                              "rules": [
                                 {
                                    "conditions": [
                                       {
                                          "fn": "booleanEquals",
                                          "argv": [
                                             true,
                                             {
                                                "fn": "getAttr",

"argv": [

                                                {
                                                   "ref": "PartitionResult"
```

```
                    },
                    "supportsFIPS"
                  ]
                }
              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                true,
                {
                  "fn": "getAttr",
                  "argv": [
                    {
                      "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                  ]
                }
              ]
            }
          ],
          "type": "tree",
          "rules": [
            {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ]
        },
        {
          "conditions": [],
          "error": "FIPS and DualStack are enabled, but this partition does not support one or both",
          "type": "error"
        }
      ]
    },
    {
      "conditions": [
```

```
                    {
                       "fn": "booleanEquals",
                       "argv": [
                          {
                             "ref": "UseFIPS"
                          },
                          true
                       ]
                    }
                 ],
                 "type": "tree",

      "rules": [
                    {
                       "conditions": [
                          {
                             "fn": "booleanEquals",
                             "argv": [
                                true,
                                {
                                   "fn": "getAttr",
                                   "argv": [
                                      {
                                         "ref": "PartitionResult"
                                      },
                                      "supportsFIPS"

                                   ]
                                }
                             ]
                          }
                       ],
                       "type": "tree",
                       "rules": [
                          {
                             "conditions": [],
                             "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                   "properties": {},
                                   "headers": {}

         },

                                "type": "endpoint"
                          }
                       ]
                    },
                    {
```

```
                              "conditions": [],
                              "error": "FIPS is enabled but this partition does not support FIPS",
                              "type": "error"
                        }
                  ]
            },
            {
                  "conditions": [
                        {
                              "fn": "booleanEquals",
                              "argv": [
                                    {
                                          "ref": "UseDualStack"
                                    },
                                    true
                              ]
                        }
                  ],
                  "type": "tree",
                  "rules": [
                        {
                              "conditions": [
                                    {
                                          "fn": "booleanEquals",
                                          "argv": [
                                                true,
                                                {

                                                "fn": "getAttr",
                                                      "argv": [
                                                            {
                                                                  "ref": "PartitionResult"
                                                            },
                                                            "supportsDualStack"
                                                      ]
                                                }
                                          ]
                                    }
                              ],
                              "type": "tree",
                              "rules": [
                                    {

                                          "conditions": [],
                                          "endpoint": {
                                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                "properties": {},
```

```
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        },
        {
            "conditions": [],
            "error": "DualStack is enabled but this partition does
not support DualStack",
            "type": "error"
        }
    ]
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
            ]
        }
    ]
},
{
    "conditions":
[],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
```

```
                "UseDualStack": true
            }
        },
        {
            "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                }
            },

        "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
            "expect": {
```

          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-

1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
        },

  "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1

with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },

    "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
```

```
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
              }
            },
      "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
            "expect":
{
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack":
true
            }
          },
```

```
                    {
                        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                            }
                        },
                        "params": {
                            "Region": "us-iso-east-1",
                            "UseFIPS": false,
                            "UseDualStack": false
                        }
                    },
                    {
                        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                        "expect": {
                            "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
                        },
                        "params": {
                            "Region": "us-isob-east-1",
                            "UseFIPS": true,
                            "UseDualStack": true
                        }
                    },
                    {
                        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                        "expect": {
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                            }
                        },
                        "params": {
                            "Region": "us-isob-east-1",
                            "UseFIPS": true,
                            "UseDualStack": false
                        }

            },
                    {
                        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
                        "expect": {
                            "error": "DualStack is enabled but this partition does not support DualStack"
                        },
                        "params": {
                            "Region": "us-isob-east-1",
                            "UseFIPS": false,
                            "UseDualStack": true
```

```json
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://example.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://example.com"
                    }
                },
                "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                "expect": {
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
```

```
                "params": {
                   "Region": "us-east-1",
                    "UseFIPS": true,
                   "UseDualStack": false,
                   "Endpoint": "https://example.com"
                 }
             },
             {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                   "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                   "Region": "us-east-1",
                   "UseFIPS": false,
                   "UseDualStack": true,
                   "Endpoint": "https://example.com"
                 }
             },
             {
                "documentation": "Missing region",
                "expect": {
                    "error": "Invalid Configuration: Missing Region"
                }
             }
          ],
          "version": "1.0"
       }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
       "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
       "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
       {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
       },
       {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
       },
       {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/ListIdentityProviders"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "IdentityProviderSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
```

```
{
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
```

```
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
    "type":
"structure",
    "members": {
      "InstanceSummaries": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
```

```
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListProductSubscriptions"
```

```
          },
          "smithy.api#paginated": {
             "inputToken": "NextToken",
             "outputToken": "NextToken",
             "pageSize": "MaxResults",
             "items": "ProductUserSummaries"
          }
       }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
       "type": "structure",
       "members": {
          "Product": {
             "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
                "smithy.api#required": {}
             }
          },
          "IdentityProvider": {
             "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
             }
          },
          "MaxResults": {
             "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
             "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
             }
          },
          "Filters": {
             "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
             "traits": {

   "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n    more sets of key-value pairs that you specify.</p>"
             }
          },
          "NextToken": {
             "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
             }
          }
       }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },

    "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated":
{
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
```

```
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {

            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }

    },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a product for a user.</p>",
          "smithy.api#required": {}
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the user.</p>"
        }
      },
      "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
        }
      },
      "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
```

```
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                 "smithy.api#documentation": "<p>An object that specifies details for the identity
    provider.</p>",
                 "smithy.api#required": {}
              }
           },
           "Product": {
              "target": "smithy.api#String",
              "traits": {
                 "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                 "smithy.api#required": {}
              }
           },
           "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                 "smithy.api#documentation": "<p>The registered identity providers product related configuration
    settings such as the\n      subnets to provision VPC endpoints.</p>"
              }
           }
        }
     },
     "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
           "IdentityProviderSummary":
    {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                 "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
    operation.</p>",
                 "smithy.api#required": {}
              }
           }
        }
     },
     "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
           "message": {
              "target": "smithy.api#String"
           }
        },
        "traits": {
           "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
           "smithy.api#error": "client",
           "smithy.api#httpError": 404
```

```
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
            "type": "string",

    "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
                "smithy.api#error": "client"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Settings": {
            "type": "structure",
            "members": {
                "Subnets": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                    "traits": {
                        "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
                        "smithy.api#length": {
                            "min": 1
                        },
                        "smithy.api#required": {}
                    }
                },
                "SecurityGroupId": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                    "traits": {
                        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n     your VPC and the VPC endpoint for activation servers.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
```

```
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n      subnets to provision VPC endpoints, and the security group ID that
is associated with the VPC\n      endpoints. The security group should permit inbound TCP port 1688
communication from resources\n      in the VPC.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related costs
will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n        monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
            "smithy.api#http": {
                "code": 200,
```

```
            "method": "POST",
            "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
```

```
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

    },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/user/StopProductSubscription"
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
            "type": "structure",
            "members": {
                "Username": {
```

```
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required":
{}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation":
"<p>Metadata that describes the start product subscription operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
```

```
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {

      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity\n     provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n     update any combination of settings in a single operation such as the:</p>\n
<ul>\n        <li>\n              <p>Subnets which you want to add to provision VPC endpoints.</p>\n          </li>\n
   <li>\n              <p>Subnets which you want to remove the VPC endpoints from.</p>\n          </li>\n
<li>\n              <p>Security group ID which permits traffic to the VPC endpoints.</p>\n          </li>\n        </ul>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#required": {}
                }
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
          "type": "structure",
          "members": {
            "AddSubnets": {

              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                  "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n     require connectivity to activation servers.</p>",
                  "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
              }
            },
            "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                "smithy.api#documentation": "<p>A security group ID
that allows inbound TCP port 1688 communication between resources in\n     your VPC and the VPC endpoints for
activation servers.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as\n     the subnets to provision VPC endpoints.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
          }
        }
```

```
      }
    }
    {
      "smithy": "2.0",
      "metadata": {
        "suppressions": [
          {
            "id": "HttpMethodSemantics",
            "namespace": "*"
          },
          {
            "id": "HttpResponseCodeSemantics",
            "namespace": "*"
          },
          {
            "id": "PaginatedTrait",
            "namespace": "*"
          },
          {
            "id": "HttpHeaderTrait",
            "namespace": "*"
          },
          {
            "id": "HttpUriConflict",
            "namespace": "*"
          },
          {
            "id": "Service",
            "namespace": "*"
          }
        ]
      },
      "shapes": {
        "com.amazonaws.licensemanager#AWSLicenseManager": {
          "type": "service",
          "version": "2018-08-01",
          "operations": [
            {
              "target": "com.amazonaws.licensemanager#AcceptGrant"
            },

          {
              "target": "com.amazonaws.licensemanager#CheckInLicense"
            },
            {
              "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
            },
            {
```

```
      "target": "com.amazonaws.licensemanager#CheckoutLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateGrantVersion"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateToken"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
      "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    }
},
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
```

```
{
  "target": "com.amazonaws.licensemanager#GetGrant"
},
{
  "target": "com.amazonaws.licensemanager#GetLicense"
},
{
  "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
},
{
  "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
},
{
  "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
},
{
  "target": "com.amazonaws.licensemanager#GetLicenseUsage"
},
{
  "target": "com.amazonaws.licensemanager#GetServiceSettings"
},
{
  "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
},
{
  "target": "com.amazonaws.licensemanager#ListDistributedGrants"
},
{
  "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
},
{
  "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
},
{
  "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
},
{
  "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
},
{
  "target": "com.amazonaws.licensemanager#ListLicenses"
},
{
  "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
},
{
```

```
      "target": "com.amazonaws.licensemanager#ListLicenseVersions"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListResourceInventory"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTokens"
    },
    {
      "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
     },
    {
      "target": "com.amazonaws.licensemanager#RejectGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
    }
  ],
```

```
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager",
        "arnNamespace": "license-manager",
        "cloudFormationName": "LicenseManager",
        "cloudTrailEventSource": "licensemanager.amazonaws.com",
        "endpointPrefix": "license-manager"
    },
    "aws.auth#sigv4": {
        "name": "license-manager"
    },
    "aws.protocols#awsJson1_1": {},
    "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n        Amazon Web Services accounts and on-premises servers.</p>",
    "smithy.api#title": "AWS License Manager",
    "smithy.api#xmlNamespace": {
        "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
    },
    "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
            "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "Boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
                "type": "Boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
```

```
        "type": "String"
      }
    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]

          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }
            ],
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
            "type": "error"
          },
          {

            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
            "type": "error"
```

```
          },
          {
            "conditions": [],
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },
              "properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ]
      },
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "type": "tree",
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref":
```

```
      "UseFIPS"
                            },
                            true
                         ]
                      },
                      {
                         "fn": "booleanEquals",
                         "argv": [
                            {
                               "ref": "UseDualStack"
                            },
                            true
                         ]
                      }
                   ],
                   "type": "tree",
                   "rules": [
                      {
                         "conditions": [
                            {
                               "fn": "booleanEquals",
                               "argv": [
                                  true,
                                  {
                                     "fn": "getAttr",
                                     "argv": [
                                        {
                                           "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                     ]

                               }
                            ]
                         },
                         {
                            "fn": "booleanEquals",
                            "argv": [
                               true,
                               {
                                  "fn": "getAttr",
                                  "argv": [
                                     {
                                        "ref": "PartitionResult"
                                     },

      "supportsDualStack"
                                  ]
```

```
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [],
                  "endpoint": {
                    "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}

                  },
                  "type": "endpoint"
                }
              ]
            },
            {
              "conditions": [],
              "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
              "type": "error"
            }
          ]
        },
        {
          "conditions": [
            {
              "fn": "booleanEquals",

        "argv": [
                {
                  "ref": "UseFIPS"
                },
                true
              ]
            }
          ],
          "type": "tree",
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
```

```
                        {
                            "fn": "getAttr",
                            "argv": [
                              {
                                 "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                         }
                       ]
                    }
                 ],
                 "type": "tree",
                 "rules": [

                    {
                         "conditions": [],
                         "endpoint": {
                            "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                         },
                         "type": "endpoint"
                       }
                    ]
                 },
                 {
                    "conditions": [],
                    "error": "FIPS is enabled but this
partition does not support FIPS",
                    "type": "error"
                 }
               ]
            },
            {
               "conditions": [
                  {
                     "fn": "booleanEquals",
                     "argv": [
                        {
                           "ref": "UseDualStack"
                        },
                        true
                      ]
                   }
                ],
               "type": "tree",
```

```
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },

              "supportsDualStack"

            ]
          }
        ]
      }
    ],
    "type": "tree",
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}

        },
        "type": "endpoint"
      }
    ]
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
]
},
{
  "conditions": [],
  "endpoint": {
    "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
```

```
            },
              "type": "endpoint"
            }
          ]
        }
      ]
    },
    {
      "conditions": [],
      "error": "Invalid Configuration: Missing Region",
      "type": "error"
    }
  ]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint":

{

            "url": "https://license-manager.af-south-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "af-south-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-1.amazonaws.com"
```

```
        }
      },
      "params": {
        "Region": "ap-northeast-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-2.amazonaws.com"
        }
      },
      "params": {
        "Region": "ap-northeast-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-3.amazonaws.com"
        }
      },
      "params": {
        "Region": "ap-northeast-3",
        "UseFIPS": false,
        "UseDualStack": false
      }

    },
    {
      "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-south-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "ap-south-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
```

```
        {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-1.amazonaws.com"

}
        },
        "params": {
          "Region": "ap-southeast-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-southeast-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-southeast-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {

"documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-southeast-3.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-southeast-3",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ca-central-1.amazonaws.com"
          }
        },
```

```
              "params": {
                  "Region": "ca-central-1",
                  "UseFIPS": false,
                  "UseDualStack": false
              }
          },
          {
              "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
              "expect": {
                  "endpoint": {
                      "url": "https://license-manager.eu-central-1.amazonaws.com"
                  }
              },
              "params": {
                  "Region": "eu-central-1",
                  "UseFIPS": false,
                  "UseDualStack": false
              }
          },
          {
              "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
              "expect": {
                  "endpoint": {
                      "url": "https://license-manager.eu-north-1.amazonaws.com"
                  }
              },
              "params": {
                  "Region": "eu-north-1",
                  "UseFIPS": false,
                  "UseDualStack": false
              }
          },
          {
              "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
              "expect": {
                  "endpoint": {
                      "url": "https://license-manager.eu-south-1.amazonaws.com"
                  }
              },
              "params": {
                  "Region":
"eu-south-1",
                  "UseFIPS": false,
                  "UseDualStack": false
              }
          },
```

```
              {
                "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager.eu-west-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "eu-west-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
                "expect": {

                  "endpoint": {
                      "url": "https://license-manager.eu-west-2.amazonaws.com"
                    }
                },
                "params": {
                  "Region": "eu-west-2",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager.eu-west-3.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "eu-west-3",
                  "UseFIPS": false,

"UseDualStack": false
                }
              },
              {
                "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager.me-south-1.amazonaws.com"
                  }
                },
```

```
        "params": {
          "Region": "me-south-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.sa-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "sa-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }

      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
```

```
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-east-2.amazonaws.com"
          }
        }
      },
      "params": {
        "Region": "us-east-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-east-2.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-2",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region
us-west-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-west-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-west-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-west-1.amazonaws.com"
        }
      },
      "params": {
```

```
                "Region": "us-west-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-west-2.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-west-2",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

"expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-west-2.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-west-2",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,

                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-northwest-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    }
  }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
```

```
          "UseFIPS": true,
          "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },

    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
```

```
              "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-gov-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-west-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-west-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-gov-east-1.api.aws"
            }
          },

      "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": true,
              "UseDualStack": true
```

```
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-gov-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
               "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
```

```
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled
but this partition does not support DualStack"
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },

        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
            }
```

```
        },

    "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

    "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,

    "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
```

          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "Missing region",
            "expect": {
              "error": "Invalid Configuration: Missing Region"
            }
          }
        ],
        "version": "1.0"
      }
    }
  },
  "com.amazonaws.licensemanager#AcceptGrant": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

},
    "output": {
      "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
    },
    "errors": [
      {

```
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#AcceptGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant
```

```
ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#AccessDeniedException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {

  "code": "ServiceAccessDenied",
          "httpResponseCode": 401
        },
        "smithy.api#documentation": "<p>Access to resource denied.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 401
      }
    },
    "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
      "type": "enum",
      "members": {
        "DISTRIBUTED_GRANTS_ONLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
          }
        },
        "ALL_GRANTS_PERMITTED_BY_ISSUER": {
```

```json
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#AllowedOperation": {

      "type": "enum",
      "members": {
        "CREATE_GRANT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateGrant"
          }
        },
        "CHECKOUT_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutLicense"
          }
        },
        "CHECKOUT_BORROW_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutBorrowLicense"
          }
        },
        "CHECK_IN_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckInLicense"
          }
        },
        "EXTEND_CONSUMPTION_LICENSE":
{
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ExtendConsumptionLicense"
          }
        },
        "LIST_PURCHASED_LICENSES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ListPurchasedLicenses"
          }
        },
```

```
      "CREATE_TOKEN": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "CreateToken"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#AllowedOperationList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#AllowedOperation"
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max":
7
    }
  }
},
  "com.amazonaws.licensemanager#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 2048
      },
      "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
    }
  },
  "com.amazonaws.licensemanager#ArnList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Arn"
    }
  },
  "com.amazonaws.licensemanager#AuthorizationException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError":
```

```
{
            "code": "AuthorizationFailure",
            "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n        policy associated with this account.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 403
    }
},
"com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
    "type": "structure",
    "members": {
        "LastRunTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
                "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes automated discovery.</p>"
    }
},

"com.amazonaws.licensemanager#Boolean": {
    "type": "boolean",
    "traits": {
        "smithy.api#default": false
    }
},
"com.amazonaws.licensemanager#BorrowConfiguration": {
    "type": "structure",
    "members": {
        "AllowEarlyCheckIn": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
                "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxTimeToLiveInMinutes": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
                "smithy.api#required": {}
            }
        }
```

```
        },
            "traits": {
                "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#BoxBoolean": {
            "type": "boolean"
        },
        "com.amazonaws.licensemanager#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanager#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanager#CheckInLicense": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
```

```
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
        }
      },
      "com.amazonaws.licensemanager#CheckInLicenseRequest":
{
          "type": "structure",
          "members": {
            "LicenseConsumptionToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License consumption token.</p>",
                "smithy.api#required": {}
              }
            },
            "Beneficiary": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License beneficiary.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
      "com.amazonaws.licensemanager#CheckInLicenseResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
      "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
          },
          "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
      },
      {
        "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
    }
  },
  "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
    "type": "structure",
    "members":
{
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
          "smithy.api#required": {}
        }
```

```
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
            "smithy.api#required": {}
          }
        },
        "DigitalSignatureMethod": {
          "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
          "traits": {
            "smithy.api#documentation": "<p>Digital signature method. The possible
value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
            "smithy.api#required": {}
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
```

```
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },
        "EntitlementsAllowed":
{
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "SignedToken": {
          "target": "com.amazonaws.licensemanager#SignedToken",
          "traits": {
            "smithy.api#documentation": "<p>Signed token.</p>"
          }
        },
        "IssuedAt": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
          }
        },
        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
        }
      },
```

```
    "traits": {
      "smithy.api#output": { }
    }
  },
  "com.amazonaws.licensemanager#CheckoutLicense": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
     },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target":
"com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n          <p>If the
account that created the license is the same that is performing the check out, you must\n          specify the account
```

as the beneficiary.</p>\n   </note>"
            }
        },
        "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
            "type": "structure",
            "members": {
                "ProductSKU": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Product
SKU.</p>",
                        "smithy.api#required": {}
                    }
                },
                "CheckoutType": {
                    "target": "com.amazonaws.licensemanager#CheckoutType",
                    "traits": {
                        "smithy.api#documentation": "<p>Checkout type.</p>",
                        "smithy.api#required": {}
                    }
                },
                "KeyFingerprint": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Entitlements": {
                    "target": "com.amazonaws.licensemanager#EntitlementDataList",
                    "traits": {
                        "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
                    }
                },
                "ClientToken": {
                    "target": "com.amazonaws.licensemanager#ClientToken",
                    "traits": {
                        "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Beneficiary": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License beneficiary.</p>"

```
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        }
      }
    },
    "traits": {

  "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
    "type": "structure",
    "members": {
      "CheckoutType": {
        "target": "com.amazonaws.licensemanager#CheckoutType",
        "traits": {
          "smithy.api#documentation": "<p>Checkout type.</p>"
        }
      },
      "LicenseConsumptionToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License consumption token.</p>"
        }
      },
      "EntitlementsAllowed": {
        "target": "com.amazonaws.licensemanager#EntitlementDataList",
        "traits": {
          "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
        }
      },
      "SignedToken": {
        "target": "com.amazonaws.licensemanager#SignedToken",
        "traits": {
          "smithy.api#documentation": "<p>Signed token.</p>"
        }
      },
      "NodeId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Node ID.</p>"
        }
      },
      "IssuedAt": {
```

```
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
                }
            },
            "Expiration": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
                }
            },
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CheckoutType": {
        "type": "enum",
        "members": {
            "PROVISIONAL": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "PROVISIONAL"
                }
            },
            "PERPETUAL": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "PERPETUAL"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#ClientRequestToken": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 36
            }
        }
```

```
    },
    "com.amazonaws.licensemanager#ClientToken": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 2048
        },
        "smithy.api#pattern": "^\\S+$"
      }
    },
    "com.amazonaws.licensemanager#ConflictException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "ConflictException",
          "httpResponseCode":
409
        },
        "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 409
      }
    },
    "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
      "type": "structure",
      "members": {
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      },
```

```
"traits": {
        "smithy.api#documentation": "<p>Details about license consumption.</p>"
    }
},
"com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
    }
},
"com.amazonaws.licensemanager#ConsumptionConfiguration": {
    "type": "structure",
    "members": {
        "RenewType": {
            "target": "com.amazonaws.licensemanager#RenewType",
            "traits": {
                "smithy.api#documentation": "<p>Renewal frequency.</p>"
            }
        },
        "ProvisionalConfiguration": {
            "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
            }
        },
        "BorrowConfiguration": {
            "target": "com.amazonaws.licensemanager#BorrowConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
    }
},
"com.amazonaws.licensemanager#CreateGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
```

```
        {
          "target":
"com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrantRequest": {
      "type": "structure",
      "members": {
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant name.</p>",
            "smithy.api#required": {}

  }
        },
        "LicenseArn": {
```

```
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
          "smithy.api#required": {}
        }
      },
      "Principals": {
        "target": "com.amazonaws.licensemanager#PrincipalArnList",
        "traits": {
          "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n          (ARN):</p>\n     <ul>\n          <li>\n              <p>An Amazon Web Services
account, which includes only the account specified.</p>\n          </li>\n     </ul>\n     <ul>\n          <li>\n
   <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n          </li>\n     </ul>\n
<ul>\n
        <li>\n            <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n     </ul>",
          "smithy.api#required": {}
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the grant.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowedOperations": {
        "target": "com.amazonaws.licensemanager#AllowedOperationList",
        "traits": {
          "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

  "com.amazonaws.licensemanager#CreateGrantResponse": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Grant ARN.</p>"
        }
      },
```

```json
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateGrantVersion":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
```

```
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
        "type": "structure",
        "members": {
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
                }
            },
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant name.</p>"
                }
            },
            "AllowedOperations": {
                "target": "com.amazonaws.licensemanager#AllowedOperationList",
                "traits": {
                    "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>"
                }
            },
            "StatusReason": {
                "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status reason.</p>"
```

```
                    }
                },
                "SourceVersion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Current version of the grant.</p>"
                    }
                },
                "Options": {
                    "target": "com.amazonaws.licensemanager#Options",
                    "traits": {
                        "smithy.api#documentation": "<p>The options specified for the grant.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant ARN.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>New version of the grant.</p>"
                    }
                }
            },
            "traits":
{
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicense": {
            "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
         },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
 all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of
licenses purchased and used.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
```

              "traits": {
                "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
              }
            },
            "LicenseCountHardLimit": {
              "target": "com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
Exceeding a hard limit\n   blocks the launch of new instances.</p>"
              }
            },
            "LicenseRules": {
              "target":
 "com.amazonaws.licensemanager#StringList",
              "traits": {
                "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n   vary by dimension, as follows.</p>\n
\n   \n   <p>\n   <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
      <code>licenseAffinityToHost</code> |\n   <code>maximumCores</code> |
<code>minimumCores</code>\n   </p>\n   \n   \n   <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n   <code>maximumCores</code> |
<code>minimumCores</code> |\n   <code>maximumSockets</code> | <code>minimumSockets</code> |\n
      <code>maximumVcpus</code> | <code>minimumVcpus</code>\n   </p>\n   \n
\n
  <p>\n   <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n   <code>maximumSockets</code> |
<code>minimumSockets</code>\n   </p>\n   \n   \n   <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n   <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n   </p>\n   \n   \n   <p>The unit for
<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for
<code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n
<code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n
<code>True</code> and <code>False</code>.</p>"
              }
            },

      "Tags": {
              "target": "com.amazonaws.licensemanager#TagList",
              "traits": {
                "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
              }
            },
            "DisassociateWhenNotFound": {
              "target": "com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"

```
              }
            },
            "ProductInformationList": {
              "target": "com.amazonaws.licensemanager#ProductInformationList",
              "traits": {
                "smithy.api#documentation": "<p>Product information.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
        {
            "type": "structure",
            "members": {
              "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
             },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {

          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
            "smithy.api#required": {}
          }
        },
        "SourceLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n        For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
            "smithy.api#required": {}
          }
        },
        "DestinationLicenseContext":
 {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
```

```
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
            "type": "structure",
            "members": {
                "LicenseConversionTaskId": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

    "traits": {
                        "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
```

```
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "ReportGeneratorName":
{
          "target": "com.amazonaws.licensemanager#ReportGeneratorName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>",
            "smithy.api#required": {}
          }
        },
        "Type": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n    <ul>\n        <li>\n            <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n        </li>\n        <li>\n            <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n        </li>\n
</ul>",
            "smithy.api#required": {}

        }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
            "smithy.api#required": {}
          }
        },
        "ReportFrequency": {
          "target": "com.amazonaws.licensemanager#ReportFrequency",
          "traits": {
            "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
            "smithy.api#required": {}
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientRequestToken",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Unique,
case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                "smithy.api#required": {}
            }
        },
        "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
                "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
    "type": "structure",
    "members": {
        "LicenseManagerReportGeneratorArn":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#CreateLicenseRequest": {
    "type": "structure",
    "members": {
        "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>License name.</p>",
                "smithy.api#required": {}
            }
```

```
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product
name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>",
            "smithy.api#required": {}
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#Issuer",
          "traits": {
            "smithy.api#documentation": "<p>License issuer.</p>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region for the license.</p>",
            "smithy.api#required": {}

    }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>",
              "smithy.api#required": {}
            }
          },
          "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n      running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n      usage.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license.</p>"
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
```

```
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License version.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#CreateLicenseVersion": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#RedirectException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
  }
```

```
        },
      "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target":
"com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product name.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits":
{
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
              "smithy.api#required": {}
            }
```

```
      },
      "LicenseMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
          "smithy.api#documentation": "<p>Information about the license.</p>"
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>",
          "smithy.api#required": {}
        }
      },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
          "smithy.api#required": {}
        }
      },

  "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
        "traits": {
          "smithy.api#documentation": "<p>License status.</p>",
          "smithy.api#required": {}
        }
      },
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientToken",
        "traits": {
          "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
          "smithy.api#required": {}
        }
      },
      "SourceVersion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Current version of the license.</p>"
        }
      }
    },
    "traits": {
```

```
"smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>License ARN.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>New version of the license.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output":
{}
      }
    },
    "com.amazonaws.licensemanager#CreateToken": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateTokenRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateTokenResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
```

```
            {
              "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },

         {
              "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n       <p>A refresh token is a JWT
token used to get an access token. With an access token,\n       you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n       call License Manager to manage the specified license.</p>"
          }
        },
        "com.amazonaws.licensemanager#CreateTokenRequest": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits":
 {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n       JWT token.</p>",
                "smithy.api#required": {}
              }
            },
            "RoleArns": {
              "target": "com.amazonaws.licensemanager#ArnList",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n       License Manager does not check whether the roles are in use.</p>"
              }
            },
            "ExpirationInDays": {
              "target": "com.amazonaws.licensemanager#Integer",
              "traits": {
                "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
              }
```

```
        },
        "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
            "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n        to the amr claim of the JWT token.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
},
"com.amazonaws.licensemanager#CreateTokenResponse": {
  "type": "structure",
  "members": {
    "TokenId": {
      "target": "com.amazonaws.licensemanager#String",

       "traits": {
          "smithy.api#documentation": "<p>Token ID.</p>"
        }
      },
      "TokenType": {
        "target": "com.amazonaws.licensemanager#TokenType",
        "traits": {
          "smithy.api#documentation": "<p>Token type.</p>"
        }
      },
      "Token": {
        "target": "com.amazonaws.licensemanager#TokenString",
        "traits": {
          "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#DateTime": {
```

```
          "type": "timestamp"
        },
        "com.amazonaws.licensemanager#DatetimeRange": {
          "type": "structure",
          "members": {
            "Begin": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Start of the time range.</p>",
                "smithy.api#required": {}
              }
            },
            "End": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>End of the time range.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
          }
        },
        "com.amazonaws.licensemanager#DeleteGrant": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
```

```
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
        }
    },
    "com.amazonaws.licensemanager#DeleteGrantRequest": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "StatusReason": {
                "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                "traits": {
                    "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Current version of the
grant.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteGrantResponse": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
```

          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation":
"<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {

```
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
        },

      "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
        "type": "structure",
        "members":
{
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```json
                    "smithy.api#documentation": "<p>ID of the license configuration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
```

```
              }
          ],
          "traits": {
              "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
          }
      },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
          "type": "structure",
          "members": {
              "LicenseManagerReportGeneratorArn": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                      "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
                      "smithy.api#required": {}
                  }
              }
          },
          "traits": {

      "smithy.api#input": {}
          }
      },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
          "type": "structure",
          "members": {},
          "traits": {
              "smithy.api#output": {}
          }
      },
      "com.amazonaws.licensemanager#DeleteLicenseRequest": {
          "type": "structure",
          "members": {
              "LicenseArn": {
                  "target": "com.amazonaws.licensemanager#Arn",
                  "traits": {
                      "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                      "smithy.api#required": {}
                  }
              },
              "SourceVersion": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                      "smithy.api#documentation": "<p>Current version of the license.</p>",
                      "smithy.api#required":
```

```
{ }
            }
          }
        },
        "traits": {
          "smithy.api#input": { }
        }
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseResponse": {
      "type": "structure",
      "members": {
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "DeletionDate": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": { }
      }
    },
    "com.amazonaws.licensemanager#DeleteToken": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },

      {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenRequest": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}

  }
    },
    "com.amazonaws.licensemanager#DigitalSignatureMethod": {
      "type": "enum",
      "members": {
        "JWT_PS384": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "JWT_PS384"
          }
```

```
        }
      }
    },
    "com.amazonaws.licensemanager#Entitlement": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement name.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#Long",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
          }
        },
        "Overage": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement unit.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowCheckIn": {
          "target":
"com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
          }
        }
      },
      "traits": {
```

```
              "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
          }
      },
      "com.amazonaws.licensemanager#EntitlementData": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement data name.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement
data value.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement data unit.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#EntitlementDataList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#EntitlementData"
        }
      },
      "com.amazonaws.licensemanager#EntitlementDataUnit": {
        "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {

"smithy.api#enumValue": "Count"
            }
          },
```

```json
        "NONE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "None"
          }
        },
        "SECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Seconds"
          }
        },
        "MICROSECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Microseconds"
          }
        },
        "MILLISECONDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Milliseconds"
          }
        },
        "BYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bytes"
          }
        },
        "KILOBYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes"
          }
        },
        "MEGABYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes"
          }
        },
        "GIGABYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes"
          }
        },
```

```
        "TERABYTES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabytes"
          }
        },
        "BITS":
{

          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits"
          }
        },
        "KILOBITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits"
          }
        },
        "MEGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits"
          }
        },
        "GIGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits"
          }
        },
        "TERABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits"

    }
        },
        "PERCENT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Percent"
          }
        },
        "BYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bytes/Second"
```

```
          }
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabytes/Second"
          }
        },
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
```

```
                  "smithy.api#enumValue": "Gigabits/Second"
                }
            },
            "TERABITS_PER_SECOND": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "Terabits/Second"
              }
            },
            "COUNT_PER_SECOND": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "Count/Second"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#EntitlementList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Entitlement"
          }
        },
        "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#EntitlementUnit": {
          "type": "enum",
          "members": {
            "COUNT": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "Count"
              }
            },
            "NONE": {
              "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue": "None"
      }
    },
    "SECONDS":
{
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Seconds"
      }
    },
    "MICROSECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Microseconds"
      }
    },
    "MILLISECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Milliseconds"
      }
    },
    "BYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
```

```json
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bits"
      }
    },
    "KILOBITS": {
      "target": "smithy.api#Unit",
      "traits": {

        "smithy.api#enumValue": "Kilobits"
      }
    },
    "MEGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabits"
      }
    },
    "GIGABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabits"
      }
    },
    "TERABITS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabits"
      }
    },
    "PERCENT": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Percent"
      }
    },
    "BYTES_PER_SECOND": {
      "target":
"smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes/Second"
      }
```

```
      },
      "KILOBYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes/Second"
        }
      },
      "MEGABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes/Second"
        }
      },
      "GIGABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes/Second"
        }
      },
      "TERABYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {

"smithy.api#enumValue": "Terabytes/Second"
        }
      },
      "BITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits/Second"
        }
      },
      "KILOBITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits/Second"
        }
      },
      "MEGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabits/Second"
        }
      },
      "GIGABITS_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits/Second"
```

```
        }
      },

    "TERABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits/Second"
          }
        },
      "COUNT_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Count/Second"
          }
        }
      }
    }
  },
  "com.amazonaws.licensemanager#EntitlementUsage": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement usage name.</p>",
          "smithy.api#required": {}
        }
      },
      "ConsumedValue": {
        "target": "com.amazonaws.licensemanager#String",

      "traits": {
          "smithy.api#documentation": "<p>Resource usage consumed.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxCount": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
        }
      },
      "Unit": {
        "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
          "smithy.api#required": {}
        }
      }
```

```
        },
      "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList":
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementUsage"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
         },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
```

```
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>",
            "smithy.api#required": {}

      }
        },
        "DryRun": {
          "target": "com.amazonaws.licensemanager#Boolean",
          "traits": {
            "smithy.api#default": false,
            "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },

        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#FailedDependencyException": {
      "type": "structure",
      "members": {
        "Message": {
```

```
          "target": "com.amazonaws.licensemanager#Message"
        },
        "ErrorCode": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "FailedDependency",
          "httpResponseCode": 424
        },
        "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 424
      }
    },
    "com.amazonaws.licensemanager#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#FilterName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
          }
        },
        "Values": {
          "target": "com.amazonaws.licensemanager#FilterValues",
          "traits": {
            "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n      describe operation. Filters can be used
to match a set of resources by specific criteria,\n      such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanager#FilterLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
```

```
            "aws.protocols#awsQueryError": {
              "code": "FilterLimitExceeded",
              "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#FilterList": {
      "type": "list",
       "member": {
          "target": "com.amazonaws.licensemanager#Filter"
       }
    },
    "com.amazonaws.licensemanager#FilterName": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#FilterValue": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#FilterValues": {
      "type": "list",
      "member": {
          "target": "com.amazonaws.licensemanager#FilterValue",
          "traits": {
             "smithy.api#xmlName": "item"
          }
      }
    },
    "com.amazonaws.licensemanager#Filters": {
      "type": "list",
      "member": {
          "target": "com.amazonaws.licensemanager#Filter",
          "traits": {
             "smithy.api#xmlName": "item"
          }
      }
    },
    "com.amazonaws.licensemanager#GetAccessToken": {
      "type": "operation",
      "input": {
          "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
      },
      "output": {
          "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
      },
      "errors": [
```

```
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
 tokens\n      are valid for one hour.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenRequest": {
          "type": "structure",
          "members": {
            "Token": {
              "target": "com.amazonaws.licensemanager#TokenString",
              "traits": {
                "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
                "smithy.api#required": {}
              }
            },
            "TokenProperties": {
              "target": "com.amazonaws.licensemanager#MaxSize3StringList",
              "traits": {
                "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
          "type": "structure",
          "members": {
```

```
      "AccessToken": {
        "target": "com.amazonaws.licensemanager#TokenString",
        "traits": {
          "smithy.api#documentation": "<p>Temporary access token.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#GetGrant": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#GetGrantRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#GetGrantResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
  }
},
"com.amazonaws.licensemanager#GetGrantRequest": {
```

```
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
           "traits": {
             "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
             "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
             "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrantResponse": {
      "type": "structure",
      "members": {
        "Grant": {
          "target": "com.amazonaws.licensemanager#Grant",
          "traits": {
             "smithy.api#documentation": "<p>Grant details.</p>"
          }
        }
      },
      "traits": {

"smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#GetLicense": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetLicenseRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetLicenseResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
```

```
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }

      },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
```

```
          }
        },
        "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits":
{
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
          "type": "structure",
          "members": {
            "LicenseConfigurationId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
              }
            },
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
              }
            },
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Name of the license configuration.</p>"
              }
            },
            "Description": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Description of the license configuration.</p>"
              }
            },
            "LicenseCountingType": {
              "target": "com.amazonaws.licensemanager#LicenseCountingType",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules.</p>"
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",


          "traits": {
            "smithy.api#documentation": "<p>Number of available licenses.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License configuration status.</p>"
          }

        },
        "OwnerAccountId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
          }
        },
        "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
          }
        },
```

```
    "ManagedResourceSummaryList": {
      "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanager#TagList",
      "traits": {
        "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
      }
    },
    "ProductInformationList": {
      "target": "com.amazonaws.licensemanager#ProductInformationList",
      "traits": {
        "smithy.api#documentation": "<p>Product information.</p>"
      }
    },
    "AutomatedDiscoveryInformation": {
      "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
      "traits": {
        "smithy.api#documentation": "<p>Automated discovery information.</p>"
      }
    },
    "DisassociateWhenNotFound": {
      "target": "com.amazonaws.licensemanager#BoxBoolean",
      "traits": {
        "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#GetLicenseConversionTask": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
```

```
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {

        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
        }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                "traits": {
                    "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                "traits": {
                    "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
                }
            },
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
```

```
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
            }
          },

    "StartTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
            }
          },
          "LicenseConversionTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
            }
          },
          "EndTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
            }
          }
        },
```

      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn": {

```
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
            "smithy.api#required":
{}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
      "type": "structure",
      "members": {
        "ReportGenerator": {
          "target": "com.amazonaws.licensemanager#ReportGenerator",
          "traits": {
            "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
```

```
      },
    "com.amazonaws.licensemanager#GetLicenseResponse": {
      "type": "structure",
      "members": {
        "License": {
          "target": "com.amazonaws.licensemanager#License",
          "traits": {
            "smithy.api#documentation": "<p>License details.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },

"com.amazonaws.licensemanager#GetLicenseUsage": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {

"target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
    }
```

```
      },
      "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
        "type": "structure",
        "members": {
          "LicenseUsage": {
            "target":
"com.amazonaws.licensemanager#LicenseUsage",
            "traits": {
              "smithy.api#documentation": "<p>License usage details.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
        }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "S3BucketArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"

            }
            },
            "SnsTopicArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
                }
            },
            "OrganizationConfiguration": {
                "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n        cross-account discovery.</p>"
                }
            },
            "EnableCrossAccountsDiscovery": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
                }
```

            },
          "LicenseManagerResourceShareArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n   provides member accounts with access to this share.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#Grant": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          },
          "GrantName": {

  "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant name.</p>",
              "smithy.api#required": {}
            }
          },
          "ParentArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Parent ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "GranteePrincipalArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {

"smithy.api#documentation": "<p>The grantee principal ARN.</p>",

                "smithy.api#required": {}
            }
        },
        "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                "smithy.api#required": {}
            }
        },
        "GrantStatus": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>",
                "smithy.api#required": {}
            }
        },
        "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
                "smithy.api#documentation": "<p>Grant status reason.</p>"
            }
        },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>",
                "smithy.api#required": {}
            }
        },
        "GrantedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
                "smithy.api#documentation": "<p>Granted operations.</p>",
                "smithy.api#required": {}
            }
        },
        "Options": {
            "target": "com.amazonaws.licensemanager#Options",
            "traits": {
                "smithy.api#documentation": "<p>The options specified for the grant.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation":

"<p>Describes a grant.</p>"
            }
        },
        "com.amazonaws.licensemanager#GrantList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#Grant"
            }
        },
        "com.amazonaws.licensemanager#GrantStatus": {
            "type": "enum",
            "members": {
                "PENDING_WORKFLOW": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "PENDING_WORKFLOW"
                    }
                },
                "PENDING_ACCEPT": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "PENDING_ACCEPT"
                    }
                },
                "REJECTED": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "REJECTED"

                    }
                },
                "ACTIVE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "ACTIVE"
                    }
                },
                "FAILED_WORKFLOW": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "FAILED_WORKFLOW"
                    }
                },
                "DELETED": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "DELETED"
                    }

```
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits":
{
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>"
          }

    },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
```

```
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#IssuerDetails",
        "traits": {
          "smithy.api#documentation": "<p>Granted license issuer.</p>"
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
        "traits": {
          "smithy.api#documentation": "<p>Granted license status.</p>"
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
ISO8601-UTC format.</p>"
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>"
        }
      },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
        }
      },
      "LicenseMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
```

```
                    "traits": {
                        "smithy.api#documentation": "<p>Granted license metadata.</p>"
                    }
                },
                "CreateTime": {
                    "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Version of the granted license.</p>"
                    }
                },
                "ReceivedMetadata": {
                    "target": "com.amazonaws.licensemanager#ReceivedMetadata",
                    "traits": {
                        "smithy.api#documentation": "<p>Granted license received metadata.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
            }
        },
        "com.amazonaws.licensemanager#GrantedLicenseList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#GrantedLicense"
            }
        },
        "com.amazonaws.licensemanager#ISO8601DateTime":
{
            "type": "string",
            "traits": {
                "smithy.api#length": {
                    "min": 0,
                    "max": 50
                },
                "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-
1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
            }
        },
        "com.amazonaws.licensemanager#Integer": {
            "type": "integer"
```

```
    },
    "com.amazonaws.licensemanager#InvalidParameterValueException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InvalidParameterValueProvided",
          "httpResponseCode": 400
        },
        "smithy.api#documentation":
"<p>One or more parameter values are not valid.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#InvalidResourceStateException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InvalidResourceState",
          "httpResponseCode": 400
        },
        "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },

    "com.amazonaws.licensemanager#InventoryFilter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the filter.</p>",
            "smithy.api#required": {}
          }
```

```
        },
        "Condition": {
          "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
          "traits": {
            "smithy.api#documentation": "<p>Condition of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }

      },
      "traits": {
        "smithy.api#documentation": "<p>An inventory filter.</p>"
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterCondition": {
      "type": "enum",
      "members": {
        "EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EQUALS"
          }
        },
        "NOT_EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "NOT_EQUALS"
          }
        },
        "BEGINS_WITH": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "BEGINS_WITH"
          }
        },
        "CONTAINS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CONTAINS"
          }
        }
      }
```

        },
        "com.amazonaws.licensemanager#InventoryFilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#InventoryFilter"
          }
        },
        "com.amazonaws.licensemanager#Issuer": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Issuer name.</p>",
                "smithy.api#required": {}
              }
            },
            "SignKey": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS key must have a key usage
of sign and verify, \n   and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
          }
        },
        "com.amazonaws.licensemanager#IssuerDetails": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Issuer name.</p>"
              }
            },
            "SignKey": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS key must have a key usage of sign and verify, \n   and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
              }
            },
            "KeyFingerprint": {

```
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
    }
  }
},
"com.amazonaws.licensemanager#License": {
  "type": "structure",
  "members": {
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
      }
    },
    "LicenseName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {

    "smithy.api#documentation": "<p>License name.</p>"
      }
    },
    "ProductName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product name.</p>"
      }
    },
    "ProductSKU": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product SKU.</p>"
      }
    },
    "Issuer": {
      "target": "com.amazonaws.licensemanager#IssuerDetails",
      "traits": {
        "smithy.api#documentation": "<p>License issuer.</p>"
      }
    },
    "HomeRegion": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
```

"smithy.api#documentation": "<p>Home Region of the license.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanager#LicenseStatus",
              "traits": {
                "smithy.api#documentation": "<p>License status.</p>"
              }
            },
            "Validity": {
              "target": "com.amazonaws.licensemanager#DatetimeRange",
              "traits": {
                "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
              }
            },
            "Beneficiary": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License beneficiary.</p>"
              }
            },
            "Entitlements": {

      "target": "com.amazonaws.licensemanager#EntitlementList",
              "traits": {
                "smithy.api#documentation": "<p>License entitlements.</p>"
              }
            },
            "ConsumptionConfiguration": {
              "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
              "traits": {
                "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
              }
            },
            "LicenseMetadata": {
              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                "smithy.api#documentation": "<p>License metadata.</p>"
              }
            },
            "CreateTime": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>License
creation time.</p>"
              }
            },

```
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
    }
  }
},
"com.amazonaws.licensemanager#LicenseConfiguration": {
  "type": "structure",
  "members": {
    "LicenseConfigurationId": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
      }
    },
    "LicenseConfigurationArn": {
      "target": "com.amazonaws.licensemanager#String",
       "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
      }
    },
    "Name": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Name of the license configuration.</p>"
      }
    },
    "Description": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Description of the license configuration.</p>"
      }
    },
    "LicenseCountingType": {
      "target": "com.amazonaws.licensemanager#LicenseCountingType",
      "traits": {
        "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
       }
    },
    "LicenseRules": {
      "target": "com.amazonaws.licensemanager#StringList",
      "vraits": {
        "smithy.api#documentation": "<p>License rules.</p>"
```

```
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
        }
      },
      "DisassociateWhenNotFound": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Status of the license configuration.</p>"
        }
      },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"


        }
      },
      "ConsumedLicenseSummaryList": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
        }
      },
      "ManagedResourceSummaryList": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>Product information.</p>"
          }
        },
        "AutomatedDiscoveryInformation": {

          "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
          "traits": {
            "smithy.api#documentation": "<p>Automated discovery information.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n        consumed and enforced by License Manager. Components include specifications for
the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n        must be associated with a
host), and the number of licenses purchased and used.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
      "type": "structure",
      "members": {
        "ResourceArn":

 {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Type of server resource.</p>"
          }
        },
        "ResourceOwnerId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
          }
```

              },
              "AssociationTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {

      "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
                }
              },
              "AmiAssociationScope": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
            }
          },
          "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
            "type": "list",
            "member": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
            }
          },
          "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
            "type": "enum",
            "members":
  {
              "AVAILABLE": {
                "target": "smithy.api#Unit",
                "traits": {
                  "smithy.api#enumValue": "AVAILABLE"
                }
              },
              "DISABLED": {
                "target": "smithy.api#Unit",
                "traits": {
                  "smithy.api#enumValue": "DISABLED"
                }
              }
            }
          },
          "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
            "type": "structure",
            "members": {
              "ResourceArn": {

```
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits":
{
          "smithy.api#documentation": "<p>Type of resource.</p>"
        }
      },
      "ResourceStatus": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Status of the resource.</p>"
        }
      },
      "ResourceOwnerId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
        }
      },
      "AssociationTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
        }
      },

    "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
```

```
            }
        },
        "com.amazonaws.licensemanager#LicenseConfigurations": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#LicenseConfiguration"
            }
        },
        "com.amazonaws.licensemanager#LicenseConversionContext": {

    "type": "structure",
            "members": {
                "UsageOperation": {
                    "target": "com.amazonaws.licensemanager#UsageOperation",
                    "traits": {
                        "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n        </p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
            }
        },
        "com.amazonaws.licensemanager#LicenseConversionTask": {
            "type": "structure",
            "members": {
                "LicenseConversionTaskId": {

    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                    "traits": {
                        "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
                    }
                },
                "ResourceArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
                    }
                },
                "SourceLicenseContext": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                    "traits": {
                        "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
```

```
                }
            },
            "DestinationLicenseContext": {
                "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the conversion task.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
                }
            },
            "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation":
 "<p>The time the conversion task was started at.</p>"
                }
            },
            "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskId": {
```

```
        "type": "string",
        "traits": {

      "smithy.api#length": {
            "min": 0,
            "max": 50
        },
          "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
        "type": "enum",
        "members": {
          "IN_PROGRESS": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "IN_PROGRESS"
            }
          },
          "SUCCEEDED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "SUCCEEDED"
            }
          },
          "FAILED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "FAILED"
            }
          }
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionTasks":
{
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTask"
        }
    },
    "com.amazonaws.licensemanager#LicenseCountingType": {
        "type": "enum",
        "members": {
          "VCPU": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "vCPU"
            }
```

```
        },
        "INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Instance"
          }
        },
        "CORE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Core"
          }
        },
        "SOCKET": {
          "target": "smithy.api#Unit",
           "traits": {
            "smithy.api#enumValue": "Socket"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseDeletionStatus": {
      "type": "enum",
      "members": {
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#License"
      }
    },
    "com.amazonaws.licensemanager#LicenseOperationFailure": {

      "type": "structure",
      "members": {
```

```
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Resource type.</p>"
            }
          },
          "ErrorMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Error message.</p>"
            }
          },
          "FailureTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {

    "smithy.api#documentation": "<p>Failure time.</p>"
            }
          },
          "OperationName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the operation.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
            }
          },
          "OperationRequestedBy": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
            }
          },
          "MetadataList": {

            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
```

```
              "smithy.api#documentation": "<p>Reserved.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseOperationFailureList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecification": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

              "smithy.api#required": {}
            }
          },
          "AmiAssociationScope": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecifications": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseSpecification"
        }
      },
      "com.amazonaws.licensemanager#LicenseStatus": {
        "type": "enum",
        "members": {
          "AVAILABLE": {
```

```
          "target": "smithy.api#Unit",

        "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "PENDING_AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_AVAILABLE"
          }
        },
        "DEACTIVATED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DEACTIVATED"
          }
        },
        "SUSPENDED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUSPENDED"
          }
        },
        "EXPIRED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EXPIRED"
          }
        },

      "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseUsage": {
      "type": "structure",
      "members": {
```

```
            "EntitlementUsages": {
              "target": "com.amazonaws.licensemanager#EntitlementUsageList",
              "traits": {
                "smithy.api#documentation": "<p>License entitlement usages.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

          }
        },
        "com.amazonaws.licensemanager#LicenseUsageException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "LicenseUsageFailure",
              "httpResponseCode": 412
            },
            "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 412
          }
        },
        "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
          },
          "output": {

    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
```

```
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
            }
        },
        "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
            "type": "structure",
            "members": {
                "LicenseConfigurationArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
                        "smithy.api#required": {}
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
            "type": "structure",
```

```
      "members": {
        "LicenseConfigurationAssociations": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
          "traits": {
            "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
          }
        },
        "NextToken": {

"target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrants": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
```

```
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
      "type": "structure",
      "members": {
        "GrantArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {

          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n     <ul>\n       <li>\n          <p>\n              <code>LicenseArn</code>\n          </p>\n
</li>\n       <li>\n          <p>\n              <code>GrantStatus</code>\n          </p>\n       </li>\n
<li>\n          <p>\n              <code>GranteePrincipalARN</code>\n          </p>\n       </li>\n
<li>\n          <p>\n              <code>ProductSKU</code>\n          </p>\n       </li>\n       <li>\n
<p>\n              <code>LicenseIssuerName</code>\n
       </p>\n       </li>\n     </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
      "type": "structure",
```

```
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",

      "traits": {
              "smithy.api#documentation": "<p>Distributed grant details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
        },
        "errors": [
          {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
        }
```

```
        },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
             "traits": {
               "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
               "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
        "type": "structure",
        "members": {
          "LicenseOperationFailureList": {
            "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
            "traits": {
              "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
```

```
      },
    "com.amazonaws.licensemanager#ListLicenseConfigurations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
      },

  "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license configurations
for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArns": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
```

```
                  "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
          },
          "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                  "smithy.api#documentation": "<p>Token for the next
set of results.</p>"
              }
          },
          "Filters": {
              "target": "com.amazonaws.licensemanager#Filters",
              "traits": {
                  "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n          <li>\n              <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n          Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n          Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n          <li>\n
<p>\n              <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n          Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
    </li>\n          <li>\n          <p>\n              <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n          </ul>"
              }
          }
      },
      "traits": {
          "smithy.api#input": {}
      }
  },
  "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
      "type": "structure",
      "members": {
          "LicenseConfigurations": {
              "target": "com.amazonaws.licensemanager#LicenseConfigurations",
              "traits": {
                  "smithy.api#documentation": "<p>Information about the license configurations.</p>"
              }
          },
          "NextToken": {
              "target": "com.amazonaws.licensemanager#String",

          "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
          }
      },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
      "type": "structure",
      "members": {
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {

"smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
```

```
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>\n      Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n    </p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTasks": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target":
```

"com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
            "type": "structure",
            "members": {
                "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n   \n   \n   <p>\n   <code>LicenseConfigurationArn</code>\n
</p>\n   \n   "
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits":
{
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

```
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
      "type": "structure",
      "members": {
        "ReportGenerators": {
          "target": "com.amazonaws.licensemanager#ReportGeneratorList",
          "traits": {
            "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }

        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
```

```
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
    }
  }
},
"com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanager#BoxInteger",
      "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
      }
    },
    "NextToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
  "type": "structure",
  "members": {
    "LicenseSpecifications": {
      "target": "com.amazonaws.licensemanager#LicenseSpecifications",
      "traits": {
        "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
      }
    },
```

```
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
```

```
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
             "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#LicenseList",
            "traits": {
              "smithy.api#documentation": "<p>License details.</p>"
            }
          },

"NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenses": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicensesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicensesResponse"
        },
```

```
"errors": [
    {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
        "target": "com.amazonaws.licensemanager#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
}
},
"com.amazonaws.licensemanager#ListLicensesRequest": {
    "type": "structure",
    "members": {
        "LicenseArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
            }
        },

        "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n        <p>\n                    <code>Beneficiary</code>\n        </p>\n
    </li>\n        <li>\n        <p>\n                    <code>ProductSKU</code>\n        </p>\n        </li>\n
<li>\n        <p>\n                    <code>Fingerprint</code>\n        </p>\n        </li>\n        <li>\n
<p>\n                    <code>Status</code>\n        </p>\n        </li>\n        </ul>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token
```

for the next set of results.</p>"
```
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                  "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#ListLicensesResponse": {
            "type": "structure",
            "members": {
              "Licenses": {
                "target": "com.amazonaws.licensemanager#LicenseList",
                "traits": {
                  "smithy.api#documentation": "<p>License details.</p>"
                }
              },
              "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
```
"smithy.api#documentation": "<p>Token for the next set of results.</p>"
```
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#ListReceivedGrants": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
```

```
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
         },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n        recipient as this Amazon Web Services account, your organization, or an organizational unit\n
   (OU) to which this member account belongs.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
       },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
```

```
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target":
"com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
              "smithy.api#required": {}
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
    <ul>\n         <li>\n          <p>\n            <code>ParentArn</code>\n          </p>\n        </li>\n
<li>\n          <p>\n            <code>GranteePrincipalArn</code>\n          </p>\n        </li>\n      </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

"com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
    "type": "structure",
    "members": {
      "Grants": {
        "target": "com.amazonaws.licensemanager#GrantList",
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
    "type": "structure",
    "members": {
      "GrantArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>ProductSKU</code>\n   </p>\n
     \n   \n   <p>\n   <code>LicenseIssuerName</code>\n   </p>\n
\n   \n   <p>\n   <code>LicenseArn</code>\n   </p>\n   \n
\n   <p>\n   <code>GrantStatus</code>\n   </p>\n   \n   \n
<p>\n
     <code>GranterAccountId</code>\n   </p>\n   \n   "
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },

```
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#MaxSize100",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
},
"com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
  "type": "structure",
  "members": {
    "Grants": {
      "target": "com.amazonaws.licensemanager#GrantList",
      "traits": {
        "smithy.api#documentation": "<p>Received grant details.</p>"
      }
    },
    "NextToken": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#ListReceivedLicenses": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists received licenses.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
    {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {

"target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
```

```
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n           <li>\n             <p>\n                 <code>Beneficiary</code>\n             </p>\n
</li>\n           <li>\n             <p>\n                 <code>ProductSKU</code>\n
</p>\n           </li>\n        </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#GrantedLicenseList",
            "traits": {
              "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
        "type": "structure",
        "members": {
          "LicenseArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
            }
          },

          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n                 <code>ProductSKU</code>\n            </p>\n
      </li>\n          <li>\n            <p>\n                 <code>Status</code>\n            </p>\n          </li>\n
<li>\n            <p>\n                 <code>Fingerprint</code>\n            </p>\n          </li>\n          <li>\n
<p>\n                 <code>IssuerName</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Beneficiary</code>\n            </p>\n          </li>\n       </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
        "type": "structure",
```

```
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#GrantedLicenseList",
            "traits": {
              "smithy.api#documentation": "<p>Received license details.</p>"
            }
          },

          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListResourceInventory": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FailedDependencyException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#InventoryFilterList",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n          <li>\n            <p>\n                <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
          </li>\n          <li>\n            <p>\n                <code>application_name</code> - The name of the
application. \n          Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
    </li>\n          <li>\n            <p>\n                <code>license_included</code> - The type of license included. \n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n          Possible values
are <code>sql-server-enterprise</code> | \n          <code>sql-server-standard</code> | \n          <code>sql-
server-web</code> | \n          <code>windows-server-datacenter</code>.</p>\n          </li>\n          <li>\n
    <p>\n                <code>platform</code> - The platform of the resource.\n          Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n          </li>\n          <li>\n            <p>\n
        <code>resource_id</code> - The ID of the resource. \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n          <li>\n            <p>\n
    <code>tag:<key></code> - The key/value combination of a tag assigned \n          to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n          <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n          </li>\n        </ul>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
```

```
        "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
          "type": "structure",
          "members": {
            "ResourceInventoryList": {
              "target": "com.amazonaws.licensemanager#ResourceInventoryList",
              "traits": {
                "smithy.api#documentation": "<p>Information
about the resources.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListTagsForResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
```

```
            }
        },
        "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
            "type": "structure",
            "members": {
                "ResourceArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

    "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
            "type": "structure",
            "members": {
                "Tags": {
                    "target": "com.amazonaws.licensemanager#TagList",
                    "traits": {
                        "smithy.api#documentation": "<p>Information about the tags.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListTokens": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListTokensRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#ListTokensResponse"
            },
            "errors":
[
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists your tokens.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTokensRequest": {
      "type": "structure",
      "members": {
        "TokenIds": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {

            "smithy.api#documentation": "<p>Token IDs.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n          <li>\n            <p>\n                <code>LicenseArns</code>\n            </p>\n          </li>\n
</ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
```

```
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
          "Tokens": {
            "target": "com.amazonaws.licensemanager#TokenList",
            "traits": {
              "smithy.api#documentation": "<p>Received token details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
```

```
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n        consumption details by resource at a selected point in time. Use this action to audit the\n        current
license consumption for any license inventory and configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#Filters",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n
 <li>\n        <p>\n            <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n        <p>\n
    <code>resourceAccount</code> - The ID of the account that owns the resource. \n        Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        </ul>"
          }
        }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {

"LicenseConfigurationUsageList": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Location": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Long": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#ManagedResourceSummary": {
      "type": "structure",
      "members": {
        "ResourceType":
{
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
          }
        },
        "AssociationCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
          }
        }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
        }
      },
      "com.amazonaws.licensemanager#MaxSize100": {

        "type": "integer",
        "traits": {
          "smithy.api#range": {
            "min": 1,
            "max": 100
          }
        }
      },
      "com.amazonaws.licensemanager#MaxSize3StringList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 3
          }
        }
      },
      "com.amazonaws.licensemanager#Message": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#Metadata": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The key name.</p>"
            }
          },

          "Value": {
            "target": "com.amazonaws.licensemanager#String",
```

```
              "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
          }
        },
        "com.amazonaws.licensemanager#MetadataList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Metadata"
          }
        },
        "com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#Options": {
          "type": "structure",
          "members": {
            "ActivationOverrideBehavior": {
              "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
              "traits": {
                "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n         Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n         <ul>\n            <li>\n                <p>As a license administrator, you can
optionally specify an\n                <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
</li>\n            <li>\n                <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n                account in your
organization.</p>\n            </li>\n            <li>\n                <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n                <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n             are activating the grant.</p>\n            </li>\n         </ul>\n         <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n            <dd>\n                <p>Use this value to activate a grant
without replacing any member accounts\n                active grants for the same product.</p>\n            </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n            <dd>\n                <p>Use this value to activate a
```

grant and disable other active grants in any\n   member accounts
 for the same product. This action will also replace their\n   previously activated grants with this activated
grant.</p>\n   </dd>\n   </dl>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n   override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager in the <i>License Manager User Guide</i>.</p>"
        }
    },
    "com.amazonaws.licensemanager#OrganizationConfiguration": {
        "type": "structure",
        "members": {
            "EnableIntegration": {
                "target": "com.amazonaws.licensemanager#Boolean",
                "traits": {
                    "smithy.api#default": false,
                    "smithy.api#documentation":
"<p>Enables Organizations integration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
        }
    },
    "com.amazonaws.licensemanager#PrincipalArnList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#Arn"
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 1
            }
        }
    },
    "com.amazonaws.licensemanager#ProductInformation": {
        "type": "structure",
        "members": {
            "ResourceType": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {

```
                    "smithy.api#documentation": "<p>Resource type. The
 possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductInformationFilterList": {
                "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
                "traits": {
                    "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n        <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n        <p>The following filters and are supported when
the resource type \n        is <code>SSM_MANAGED</code>:</p>\n
        <ul>\n           <li>\n              <p>\n                 <code>Application Name</code> - The name of the
application.\n         Logical operator is <code>EQUALS</code>.</p>\n           </li>\n           <li>\n
<p>\n                 <code>Application Publisher</code> - The publisher of the application.\n           Logical
operator is <code>EQUALS</code>.</p>\n           </li>\n           <li>\n              <p>\n
<code>Application Version</code> - The version of the application.\n           Logical operator is
<code>EQUALS</code>.</p>\n           </li>\n           <li>\n              <p>\n                 <code>Platform
Name</code> - The name of the platform.\n           Logical operator is <code>EQUALS</code>.</p>\n
</li>\n           <li>\n              <p>\n                 <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
        </li>\n           <li>\n              <p>\n                 <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>.  The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n              </p>\n           </li>\n           <li>\n              <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n           Logical operator is <code>NOT_EQUALS</code>.</p>\n           </li>\n
<li>\n              <p>\n                 <code>License Included</code> - The type of license included.\n           Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n           Possible values are:
 <code>sql-server-enterprise</code> | \n              <code>sql-server-standard</code> | \n              <code>sql-server-
web</code> | \n              <code>windows-server-datacenter</code>.</p>\n           </li>\n        </ul>\n
<p>The following filters and logical operators are supported when the resource type\n        is
<code>RDS</code>:</p>\n        <ul>\n           <li>\n              <p>\n                 <code>Engine Edition</code> -
The edition of the database engine.\n           Logical operator is <code>EQUALS</code>.\n           Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n           </li>\n           <li>\n              <p>\n                 <code>License Pack</code> - The license
pack.\n           Logical operator is <code>EQUALS</code>.\n           Possible values are: <code>data
guard</code> | \n              <code>diagnostic pack sqlt</code> | \n
        <code>tuning pack sqlt</code> | \n              <code>ols</code> | \n              <code>olap</code>.</p>\n
   </li>\n        </ul>",
                    "smithy.api#required": {}
                }
            }
        },
```

```
      "traits": {
        "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilter": {
      "type": "structure",
      "members": {
        "ProductInformationFilterName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Filter name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductInformationFilterValue": {
          "target": "com.amazonaws.licensemanager#StringList",


       "traits": {
            "smithy.api#documentation": "<p>Filter value.</p>"
          }
        },
        "ProductInformationFilterComparator": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Logical operator.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes product information filters.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformationFilter"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformation"
      }
    },
    "com.amazonaws.licensemanager#ProvisionalConfiguration": {
      "type": "structure",
      "members": {
```

```
          "MaxTimeToLiveInMinutes": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#RateLimitExceededException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "RateLimitExceeded",
            "httpResponseCode": 429
          },
          "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 429
        }
      },
      "com.amazonaws.licensemanager#ReceivedMetadata": {
        "type": "structure",
        "members": {
          "ReceivedStatus": {
            "target": "com.amazonaws.licensemanager#ReceivedStatus",
            "traits": {
              "smithy.api#documentation": "<p>Received status.</p>"
            }
          },
          "ReceivedStatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>Received
status reason.</p>"
            }
          },
          "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Allowed operations.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
        }
      },
      "com.amazonaws.licensemanager#ReceivedStatus": {
        "type": "enum",
        "members": {
          "PENDING_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_WORKFLOW"
            }
          },
          "PENDING_ACCEPT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue":
"PENDING_ACCEPT"
            }
          },
          "REJECTED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "REJECTED"
            }
          },
          "ACTIVE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ACTIVE"
            }
          },
          "FAILED_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED_WORKFLOW"
            }
          },
          "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DELETED"
            }
```

```
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
           "traits": {
             "smithy.api#enumValue": "DISABLED"
           }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
             "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    }
  },
  "com.amazonaws.licensemanager#RedirectException": {
    "type": "structure",
    "members": {
      "Location": {
        "target": "com.amazonaws.licensemanager#Location",
        "traits": {
           "smithy.api#httpHeader": "Location"
        }
      },
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
       "smithy.api#error": "client",
      "smithy.api#httpError": 308
    }
  },
  "com.amazonaws.licensemanager#RejectGrant": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#RejectGrantRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#RejectGrantResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#RejectGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}

  }
    },
      "com.amazonaws.licensemanager#RejectGrantResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
```

```
              "target": "com.amazonaws.licensemanager#GrantStatus",
              "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#RenewType":
{
            "type": "enum",
            "members": {
              "NONE": {
                "target": "smithy.api#Unit",
                "traits": {
                  "smithy.api#enumValue": "None"
                }
              },
              "WEEKLY": {
                "target": "smithy.api#Unit",
                "traits": {
                  "smithy.api#enumValue": "Weekly"
                }
              },
              "MONTHLY": {
                "target": "smithy.api#Unit",
                "traits": {
                  "smithy.api#enumValue": "Monthly"
                }
              }
            }
          },
        "com.amazonaws.licensemanager#ReportContext": {
          "type": "structure",
          "members": {
            "licenseConfigurationArns": {
              "target": "com.amazonaws.licensemanager#ArnList",
              "traits": {
```

```
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
    }
},
"com.amazonaws.licensemanager#ReportFrequency": {
    "type": "structure",
    "members": {
        "value": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
                "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
            }
        },
        "period": {
            "target": "com.amazonaws.licensemanager#ReportFrequencyType",

            "traits": {
                "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
    }
},
"com.amazonaws.licensemanager#ReportFrequencyType": {
    "type": "enum",
    "members": {
        "DAY": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "DAY"
            }
        },
        "WEEK": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "WEEK"
            }
        },
```

```
                    "MONTH": {
                       "target": "smithy.api#Unit",
                       "traits":
{
                          "smithy.api#enumValue": "MONTH"
                       }
                    }
                 }
              },
           "com.amazonaws.licensemanager#ReportGenerator": {
              "type": "structure",
              "members": {
                 "ReportGeneratorName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                       "smithy.api#documentation": "<p>Name of the report generator.</p>"
                    }
                 },
                 "ReportType": {
                    "target": "com.amazonaws.licensemanager#ReportTypeList",
                    "traits": {
                       "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
                    }
                 },
                 "ReportContext": {
                    "target": "com.amazonaws.licensemanager#ReportContext",
                    "traits": {
                       "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
                    }
                 },
                 "ReportFrequency": {
                    "target": "com.amazonaws.licensemanager#ReportFrequency",
                    "traits": {
                       "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
                    }
                 },
                 "LicenseManagerReportGeneratorArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                       "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
                    }
                 },
                 "LastRunStatus": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                       "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
                    }
```

                },
                "LastRunFailureReason":
{
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
                    }
                },
                "LastReportGenerationTime": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
                    }
                },
                "ReportCreatorAccount": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
                    }
                },
                "Description": {
                    "target": "com.amazonaws.licensemanager#String",

        "traits": {
                        "smithy.api#documentation": "<p>Description of the report generator.</p>"
                    }
                },
                "S3Location": {
                    "target": "com.amazonaws.licensemanager#S3Location",
                    "traits": {
                        "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
                    }
                },
                "CreateTime": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Time the report was created.</p>"
                    }
                },
                "Tags": {
                    "target": "com.amazonaws.licensemanager#TagList",
                    "traits": {
                        "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
                    }

```
          }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportGenerator"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorName": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "com.amazonaws.licensemanager#ReportType": {
      "type": "enum",
      "members": {
        "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
          }

      },
        "LICENSE_CONFIGURATION_USAGE_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationUsageReport"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportTypeList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportType"
      }
    },
    "com.amazonaws.licensemanager#ResourceInventory": {
      "type": "structure",
      "members": {
```

```
      "ResourceId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits":
{
          "smithy.api#documentation": "<p>Type of resource.</p>"
        }
      },
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "Platform": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Platform of the resource.</p>"
        }
      },
      "PlatformVersion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
        }
      },
      "ResourceOwningAccountId":
{
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about a resource.</p>"
    }
  },
  "com.amazonaws.licensemanager#ResourceInventoryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ResourceInventory"
    }
```

```
        },
        "com.amazonaws.licensemanager#ResourceLimitExceededException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "ResourceLimitExceeded",

              "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#ResourceNotFoundException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "InvalidResource.NotFound",
              "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>The resource cannot be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#ResourceType": {
          "type": "enum",
          "members": {
            "EC2_INSTANCE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "EC2_INSTANCE"
              }
            },
            "EC2_HOST": {
              "target": "smithy.api#Unit",
              "traits": {
```

```
              "smithy.api#enumValue": "EC2_HOST"
            }
          },
          "EC2_AMI": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "EC2_AMI"
            }
          },
          "RDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "RDS"
            }
          },
          "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#S3Location": {
        "type": "structure",
        "members": {
          "bucket": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
            }
          },
          "keyPrefix": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
        }
      },
      "com.amazonaws.licensemanager#ServerInternalException": {
        "type": "structure",
        "members": {
          "Message": {
```

```
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "InternalError",
          "httpResponseCode": 500
        },
        "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licensemanager#SignedToken": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 4096
        }
      }
    },
    "com.amazonaws.licensemanager#StatusReasonMessage": {
      "type": "string",
      "traits":
{
        "smithy.api#length": {
          "min": 0,
          "max": 400
        },
        "smithy.api#pattern": "^[\\s\\S]+$"
      }
    },
    "com.amazonaws.licensemanager#String": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#StringList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#String"
      }
    },
    "com.amazonaws.licensemanager#Tag": {
      "type": "structure",
      "members": {
        "Key": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Tag key.</p>"
```

```
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation":
"<p>Tag value.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagKeyList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "com.amazonaws.licensemanager#TagList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Tag"
        }
      },
      "com.amazonaws.licensemanager#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#TagResourceResponse"
        },
        "errors": [
          {
            "target":
"com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
```

```
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits":
{
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>One or more tags.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#TokenData": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>"
            }
          },
```

```
      "TokenType": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
        }
      },
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
        }
      },
      "ExpirationTime": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
 format.</p>"
        }
      },
      "TokenProperties": {
        "target": "com.amazonaws.licensemanager#MaxSize3StringList",
        "traits": {
          "smithy.api#documentation": "<p>Data specified by the caller.</p>"
        }
      },
      "RoleArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
        }
      }
    },
    "traits": {

      "smithy.api#documentation": "<p>Describes a token.</p>"
    }
  },
  "com.amazonaws.licensemanager#TokenList": {
    "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanager#TokenData"
        }
      },
      "com.amazonaws.licensemanager#TokenString": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 4096
          },
          "smithy.api#pattern": "^\\S+$"
        }
      },
      "com.amazonaws.licensemanager#TokenType": {
        "type": "enum",
        "members": {
          "REFRESH_TOKEN": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "REFRESH_TOKEN"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
{
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UntagResourceResponse"
        },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanager#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UntagResourceResponse": {
      "type": "structure",
      "members": {},
```

```
        "traits": {
          "smithy.api#output": { }
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
        },
        "output": {

  "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": { }
            }
          },
```

```
"LicenseConfigurationStatus": {
  "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
  "traits": {
    "smithy.api#documentation": "<p>New status of the license configuration.</p>"
  }
},
"LicenseRules": {
  "target": "com.amazonaws.licensemanager#StringList",
  "traits": {

    "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n configuration is licenseAffinityToHost.</p>"
  }
},
"LicenseCount": {
  "target": "com.amazonaws.licensemanager#BoxLong",
  "traits": {
    "smithy.api#documentation": "<p>New number of licenses managed by the license configuration.</p>"
  }
},
"LicenseCountHardLimit": {
  "target": "com.amazonaws.licensemanager#BoxBoolean",
  "traits": {
    "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
  }
},
"Name": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>New name of the license configuration.</p>"
  }
},
"Description": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>New description of the license configuration.</p>"
  }
},
"ProductInformationList": {
  "target": "com.amazonaws.licensemanager#ProductInformationList",
  "traits": {
    "smithy.api#documentation": "<p>New product information.</p>"
  }
},
"DisassociateWhenNotFound": {
  "target": "com.amazonaws.licensemanager#BoxBoolean",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            }
        },
        "traits":
{
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
```

```
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
 {
            "type": "structure",
            "members": {
                "LicenseManagerReportGeneratorArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ReportGeneratorName": {
                    "target": "com.amazonaws.licensemanager#ReportGeneratorName",
                    "traits": {
                        "smithy.api#documentation": "<p>Name of the report generator.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Type": {
                    "target": "com.amazonaws.licensemanager#ReportTypeList",
                    "traits": {
                        "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n        <ul>\n        <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n        </li>\n        <li>\n        <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n        </li>\n        </ul>",
                        "smithy.api#required": {}
                    }
                },
                "ReportContext": {
                    "target": "com.amazonaws.licensemanager#ReportContext",
                    "traits": {
                        "smithy.api#documentation": "<p>The report context.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ReportFrequency": {
                    "target": "com.amazonaws.licensemanager#ReportFrequency",
```

```
                    "traits":
{

                         "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                         "smithy.api#required": {}

                    }
                },
                "ClientToken": {
                    "target": "com.amazonaws.licensemanager#ClientRequestToken",
                    "traits": {
                         "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                         "smithy.api#required": {}

                    }
                },
                "Description": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                         "smithy.api#documentation": "<p>Description of the report generator.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}

            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{

            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}

            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
```

```json
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },

      {
        "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
      },
      {
        "target": "com.amazonaws.licensemanager#LicenseUsageException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
    }
  },

  "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
          "smithy.api#required": {}
        }
      },
      "AddLicenseSpecifications": {
        "target": "com.amazonaws.licensemanager#LicenseSpecifications",
        "traits": {
          "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
        }
      },
      "RemoveLicenseSpecifications": {
        "target": "com.amazonaws.licensemanager#LicenseSpecifications",
        "traits": {
          "smithy.api#documentation":
"<p>ARNs of the license configurations to remove.</p>"
        }
```

```
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
              }
          },
          "SnsTopicArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
              }
          },
          "OrganizationConfiguration": {
              "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
              "traits": {
                  "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
              }
          },
          "EnableCrossAccountsDiscovery": {
              "target": "com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                  "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

      }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#UsageOperation": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          }
        }
      },
      "com.amazonaws.licensemanager#ValidationException": {
```

```
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

        "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        }
      }
    }
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {
```

```go
	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// Current version
	of the license.
	//
	// This member is required.
	SourceVersion *string

	noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

	// Date when the license is deleted.
	DeletionDate *string

	// License status.
	Status types.LicenseDeletionStatus

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack,
	options); err != nil {
		return err
```

```go
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
	return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
	return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
	}
	if
	err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
	return err
	}
	if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
	return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
	middleware.Before); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
	return err
	}
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:     ServiceID,
 OperationName: "DeleteLicense",
 }
}
```

1.34 aws-aws-sdk-go-v2-service-scheduler 1.6.1

1.34.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)


// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
```

```go
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil,
 err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
```

```go
		return err
	}
	err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
	if err !=
nil {
	return err
	}
	err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
	if err != nil {
	return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
	return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
	return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
	return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
	return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack);
err != nil {
	return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
	return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
	return err
```

```
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
     return err
    }
    if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
     return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack,
    options); err != nil {
     return err
    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

name: License Scan

on: [pull_request]

```
jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
    matrix:
      python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
        path: sdkbase
        ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
        path: new-ref
        fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
     - name: Set up Python ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
        python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
     - name: Run Scan code on pr ref
       run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
     # compare
     - name: License test
       run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary
 *string

 noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```go
 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
```

```
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return
err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
Region:       region,
ServiceID:     ServiceID,
OperationName: "CheckInLicense",
}
}
{
  "smithy": "2.0",
```

```
"shapes": {
  "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
    "type": "long"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
      },
      "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
          "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n     Filters can be used to match a set of resources by specific criteria.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
```

          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
        "type": "structure",
        "members": {}
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n   Organizations integration has been enabled, and
 which Regions data will be aggregated from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {

```
                        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                    }
                },
                "StatusMessage": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                    "traits": {
                        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                    }
                },
                "HomeRegions": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                    "traits": {
                        "smithy.api#documentation": "<p>The Region in which License Manager displays
 the aggregated data for Linux\n     subscriptions.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
            "type": "structure",
            "members": {
                "AmiId": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
                    }
                },
                "InstanceID": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
                    }
                },
                "InstanceType": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
                    }

            },
                "AccountID": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
                    }
                },
```

```
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the instance.</p>"
        }
      },
      "Region": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
        }
      },
      "UsageOperation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
        }
      },
      "ProductCode": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
        "traits": {
          "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
        }
      },
      "LastUpdatedTime": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
        }
      },
      "SubscriptionName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    }
  },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
    ],

    "traits": {
        "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
            "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
```

```
                    "*",
                    "content-type",
                    "x-amz-content-sha256",
                    "x-amz-user-agent",
                    "x-amzn-platform-id",
                    "x-amzn-trace-id",
                    "content-length",
                    "x-api-key",
                    "authorization",
                    "x-amz-date",

    "x-amz-security-token",
                    "Access-Control-Allow-Headers",
                    "Access-Control-Allow-Methods",
                    "Access-Control-Allow-Origin"
                ],
                "additionalExposedHeaders": [
                    "x-amzn-errortype",
                    "x-amzn-requestid",
                    "x-amzn-trace-id"
                ]
            },
            "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager Linux Subscriptions",
            "smithy.rules#endpointRuleSet": {
                "version": "1.0",
                "parameters": {
                    "Region": {
                        "builtIn": "AWS::Region",
                        "required": false,
                        "documentation": "The
 AWS region used to dispatch the request.",
                        "type": "String"
                    },
                    "UseDualStack": {
                        "builtIn": "AWS::UseDualStack",
                        "required": true,
                        "default": false,
                        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                        "type": "Boolean"
                    },
                    "UseFIPS": {
                        "builtIn": "AWS::UseFIPS",
                        "required": true,
                        "default": false,
                        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
```

configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

```
          "type": "Boolean"
        },
        "Endpoint": {
          "builtIn": "SDK::Endpoint",
          "required": false,
          "documentation": "Override the endpoint used to send this request",
          "type": "String"
        }
      },
      "rules": [
        {
          "conditions": [
            {
              "fn": "isSet",
              "argv": [
                {
                  "ref": "Endpoint"
                }
              ]
            }
          ],
          "type": "tree",
          "rules": [
            {

            "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
              "type": "error"
            },
            {
              "conditions": [],
              "type": "tree",
              "rules": [
                {
```

```
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],

        "endpoint": {
                        "url": {
                            "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        }
    ]
},
{
    "conditions": [],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",

"argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
```

```
{
    "conditions": [
        {
            "fn": "aws.partition",
            "argv": [
                {
                    "ref": "Region"
                }
            ],

"assign": "PartitionResult"
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                },

{
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {

"fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
```

```
                                      "argv": [
                                        {
                                          "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                      ]

        }
                                  ]
                                },
                                {
                                  "fn": "booleanEquals",
                                  "argv": [
                                    true,
                                    {
                                      "fn": "getAttr",
                                      "argv": [
                                        {
                                          "ref": "PartitionResult"
                                        },

                                   "supportsDualStack"
                                      ]
                                    }
                                  ]
                                }
                              ],
                              "type": "tree",
                              "rules": [
                                {
                                  "conditions": [],
                                  "type": "tree",
                                  "rules": [
                                    {

                      "conditions": [],
                                      "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                      },
                                      "type": "endpoint"
                                    }
                                  ]
                                }
                              ]
```

```
                        },
                          {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                            "type": "error"
                          }
                        ]
                      },
                      {
                        "conditions": [
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              {
                                "ref": "UseFIPS"
                              },
                              true
                            ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
                          {
                            "conditions": [
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  true,

    {
                                    "fn": "getAttr",
                                    "argv": [
                                      {
                                        "ref": "PartitionResult"
                                      },
                                      "supportsFIPS"
                                    ]
                                  }
                                ]
                              }
                            ],
                            "type": "tree",

                    "rules": [
                              {
                                "conditions": [],
```

```
                                            "type": "tree",
                                            "rules": [
                                              {
                                                "conditions": [],
                                                "endpoint": {
                                                  "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                                  "properties": {},
                                                  "headers": {}

                                                },
                                                "type": "endpoint"
                                              }
                                            ]
                                          }
                                        ]
                                    },
                                    {
                                      "conditions": [],
                                      "error": "FIPS is enabled but this partition does not support FIPS",
                                      "type": "error"
                                    }
                                  ]

                              },
                                          {
                                            "conditions": [
                                              {
                                                "fn": "booleanEquals",
                                                "argv": [
                                                  {
                                                    "ref": "UseDualStack"
                                                  },
                                                  true
                                                ]
                                              }
                                            ],
                                            "type": "tree",
                                            "rules": [
                                              {

                                             "conditions": [
                                                    {
                                                      "fn": "booleanEquals",
                                                      "argv": [
                                                        true,
                                                        {
                                                          "fn": "getAttr",
```

```
                                              "argv": [
                                                {
                                                   "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                              ]
                                           }
                                        ]
                                     }
                                  ],
                                  "type": "tree",
                                  "rules": [
                                     {
                                        "conditions": [],
                                        "type": "tree",
                                        "rules": [
                                           {
                                              "conditions": [],

                                           "endpoint": {
                                                   "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                   "properties": {},
                                                   "headers": {}
                                                 },
                                                 "type": "endpoint"
                                              }
                                           ]
                                        }
                                     ]
                                  },

           {
                                     "conditions": [],
                                     "error": "DualStack is enabled but this partition does not support DualStack",
                                     "type": "error"
                                  }
                               ]
                            },
                            {
                               "conditions": [],
                               "type": "tree",
                               "rules": [
                                  {
                                     "conditions": [],
                                     "endpoint": {
                                        "url":
```

"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
    ]
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {

                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false

```
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {

          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
```

        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
           "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
             "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint":

{

             "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
             "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
           "UseDualStack": true
        }
      },
      {

          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint":

{

              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
             "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": true

```
              }
            },
            {
              "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                }
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "error": "DualStack is enabled but this partition does not support DualStack"
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {

"documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
                }
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
              },
              "params": {
```

          "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",

    "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {

```
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
```

```
                  }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }

          }
            ],
            "version": "1.0"
          }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled LinuxSubscriptionsDiscovery"
          },
          {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled LinuxSubscriptionsDiscovery"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
      "type": "structure",
      "members": {
        "SourceRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "OrganizationIntegration": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
          "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
```

```
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
          "type":
"operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with commercial Linux\n       subscriptions.</p>",
            "smithy.api#http": {

      "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Instances"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
          "type": "structure",
          "members": {
```

```
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by
 the name of\n      <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n
match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n        <p>The valid names for this filter
are:</p>\n     <ul>\n        <li>\n           <p>\n             <code>AmiID</code>\n         </p>\n
</li>\n        <li>\n          <p>\n             <code>InstanceID</code>\n          </p>\n        </li>\n
<li>\n          <p>\n             <code>AccountID</code>\n          </p>\n        </li>\n        <li>\n
<p>\n            <code>Status</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>Region</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
<code>UsageOperation</code>\n          </p>\n        </li>\n        <li>\n
   <p>\n            <code>ProductCode</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
   <code>InstanceType</code>\n          </p>\n        </li>\n      </ul>\n      <p>The valid Operators for this
filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>contains</code>\n          </p>\n
</li>\n        <li>\n          <p>\n            <code>equals</code>\n          </p>\n        </li>\n        <li>\n
   <p>\n            <code>Notequal</code>\n          </p>\n        </li>\n      </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

  }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 16384
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
```

```
                    "traits": {
                        "smithy.api#documentation":
    "<p>An array that contains instance objects.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n      organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/subscription/ListLinuxSubscriptions"
                },
                "smithy.api#idempotent": {},
                "smithy.api#paginated": {
                    "inputToken": "NextToken",
                    "outputToken": "NextToken",
                    "pageSize": "MaxResults",
                    "items": "Subscriptions"
                }
```

```
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n          <code>Subscription</code>\n        </p>\n      </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n
    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n
<li>\n        <p>\n          <code>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n
  <code>Notequal</code>\n        </p>\n      </li>\n    </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>",
            "smithy.api#length": {

              "min": 1,
              "max": 16384
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                }
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},

"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
```

            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled OrganizationIntegration"
            }
          ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [

{
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
              "name": "Completed",
              "value": "Completed",
              "documentation": "Completed status"
            },
            {
              "name": "Successful",
              "value": "Successful",
              "documentation": "Successful status"
            },
            {
              "name": "Failed",
              "value": "Failed",
              "documentation": "Failed status"
            }
          ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String",
          "traits":

```
{
          "smithy.api#length": {
            "min": 1,
            "max": 100
          }
        }
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"

        }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
```

```
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux
subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
      },

      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
```

```
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                    "smithy.api#required": {}
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",

    "smithy.api#required": {}
                }
            },
            "AllowUpdate": {
                "target": "smithy.api#Boolean",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
```

```
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```go
import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
    c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //   - Beneficiary
    //   - ProductSKU
    //   - Fingerprint
    //
    //   - Status
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string
```

```go
    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if
    err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
```

```go
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
	err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}
```

```go
func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:
			ServiceID,
		OperationName: "ListLicenses",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/grafana/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
	if params == nil {
		params = &DisassociateLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*DisassociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type DisassociateLicenseInput struct {

	// The type of license to remove from the workspace.
	//
	//
	This member is required.
	LicenseType types.LicenseType
```

```go
	// The ID of the workspace to remove the Grafana Enterprise license from.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

	// A structure containing information about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack,
	options, "DisassociateLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
```

```go
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
	return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
	return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
	return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
	return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
	return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
	return err
}
if err = addClientUserAgent(stack, options); err != nil {
	return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
	return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
	return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
	return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
	return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
	return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
	return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
	return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
	return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
```

```
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
 {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
 params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
```

```go
	return out, nil
}

type
CheckoutLicenseInput struct {

	// Checkout type.
	//
	// This member is required.
	CheckoutType types.CheckoutType

	// Unique, case-sensitive identifier that you provide to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// License entitlements.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Key fingerprint identifying the license.
	//
	// This member is required.
	KeyFingerprint *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// License beneficiary.
	Beneficiary *string

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

	// Checkout type.
	CheckoutType types.CheckoutType

	// Allowed license entitlements.
	EntitlementsAllowed []types.EntitlementData
```

```go
    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource
    Name (ARN) of the checkout license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil
    {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
```

```go
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
    return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack);
    err != nil {
    return err
    }
    if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
    middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
    }
```

```go
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
	if params == nil {
		params = &CheckoutBorrowLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*CheckoutBorrowLicenseOutput)
```

```go
	out.ResultMetadata = metadata
	return out, nil
}

type CheckoutBorrowLicenseInput struct {

	// Unique, case-sensitive identifier
	that you provide to ensure the idempotency of
	// the request.
	//
	// This member is required.
	ClientToken *string

	// Digital signature method. The possible value is JSON Web Signature (JWS)
	// algorithm PS384. For more information, see RFC 7518 Digital Signature with
	// RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
	//
	// This member is required.
	DigitalSignatureMethod types.DigitalSignatureMethod

	// License entitlements. Partial checkouts are not supported.
	//
	// This member is required.
	Entitlements []types.EntitlementData

	// Amazon Resource Name (ARN) of the license. The license must use the borrow
	// consumption configuration.
	//
	// This member is required.
	LicenseArn *string

	// Information about constraints.
	CheckoutMetadata []types.Metadata

	// Node ID.
	NodeId *string

	noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

	// Information about constraints.
	CheckoutMetadata []types.Metadata

	// Allowed license entitlements.
	EntitlementsAllowed
```

```
    []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
    return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
     return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
     return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
     return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
     return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
     return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
     return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
     return
    err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
     return err
    }
    if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
     return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
    middleware.Before); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
     return err
```

```
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:      region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }
```

```go
    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide
    to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

    // Product name.
    //
```

```go
	// This member is required.
	ProductName *string

	// Product SKU.
	//
	// This member is required.
	ProductSKU *string

	// Date and time range during which the license is valid, in ISO8601-UTC format.
	//
	// This member is required.
	Validity
	*types.DatetimeRange

	// Information about the license.
	LicenseMetadata []types.Metadata

	noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
```

```go
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
		return fmt.Errorf("add
protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
		return err
```

```go
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
        return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
        return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
        return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
        return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
        return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:       region,
        ServiceID:    ServiceID,
        OperationName:
"CreateLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
```

```go
	if params == nil {
		params = &GetLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*GetLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type GetLicenseInput struct {

	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// License version.
	Version
	*string

	noSmithyDocumentSerde
}

type GetLicenseOutput struct {

	// License details.
	License *types.License

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
```

```go
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return
err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
```

```go
  if err = addOpGetLicenseValidationMiddleware(stack);
err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
```

```go
	//
	  - ProductSKU
	//   - Status
	//   - Fingerprint
	//   - IssuerName
	//   - Beneficiary
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

	// Received license details.
	Licenses []types.GrantedLicense

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
	if err != nil {
		return err
	}
```

```go
	if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack);
	err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
	middleware.Before); err != nil {
		return err
	}
```

```
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
```

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,

indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

   9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS

   APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

     http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
{
  "smithy": "2.0",
  "shapes": {

```
"com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
    "type": "structure",
    "members": {
        "DirectoryId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation":
"<p>Details about an Active Directory identity provider.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related
costs will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
    monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/AssociateUser"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The identity provider of the user.</p>",
```

```
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n     resource.</p>",
            "smithy.api#error": "server"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
          },
```

```json
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
```

```
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
```

```
            ],
          "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
          }

        },
            "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
              "type": "structure",
              "members": {
                "Username": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                  }
                },
                "InstanceId": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                  }
                },
                "IdentityProvider": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                  "traits": {
                    "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                  }
                },
                "Domain": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The domain name of the user.</p>"
                  }
                }
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
```

```
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },

  "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used
 to return more specific results from a describe\n     operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n     attributes, or IDs.</p>"
        }
      },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
```

```
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
  "<p>Details about an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>An object that details the registered identity providers product
  related configuration\n      settings such as the subnets to provision VPC endpoints.</p>",
            "smithy.api#required":
  {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "FailureMessage": {
          "target": "smithy.api#String",
```

```
              "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation":
  "<p>Describes an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
  subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status
  of an EC2 instance resource.</p>",
                "smithy.api#required": {}
              }
            },
            "Products": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
                "smithy.api#required": {}
              }
            },
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
              }
            },
```

```
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }

        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits":
 {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
```

```
                "smithy.api#required": {}
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          }

        },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
```

```
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",

    "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },

      {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
```

```
                "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
            },
            "aws.auth#sigv4": {
                "name": "license-manager-user-subscriptions"
            },
            "aws.protocols#restJson1": {},
            "smithy.api#cors": {
                "additionalAllowedHeaders": [
                    "*",
                    "content-type",
                    "x-amz-content-sha256",
                    "x-amz-user-agent",
                    "x-amzn-platform-id",
                    "x-amzn-trace-id",
                    "content-length",
                    "x-api-key",
                    "authorization",
                    "x-amz-date",
                    "x-amz-security-token",
                    "Access-Control-Allow-Headers",
                    "Access-Control-Allow-Methods",
                    "Access-Control-Allow-Origin"
                ],
                "additionalExposedHeaders": [
                    "x-amzn-errortype",
                    "x-amzn-requestid",
                    "x-amzn-trace-id"

                ]
            },
            "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager User Subscriptions",
            "smithy.rules#endpointRuleSet": {
                "version": "1.0",
                "parameters": {
                    "Region": {
                        "builtIn": "AWS::Region",
                        "required": false,
                        "documentation": "The AWS region used to dispatch the request.",
                        "type": "String"
                    },
                    "UseDualStack": {
                        "builtIn": "AWS::UseDualStack",
                        "required": true,
                        "default": false,
                        "documentation": "When true, use the dual-stack
```

endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
      },
     "UseFIPS": {
       "builtIn": "AWS::UseFIPS",
       "required": true,
       "default": false,
       "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
       "type": "Boolean"
      },
     "Endpoint": {
       "builtIn": "SDK::Endpoint",
       "required": false,
       "documentation": "Override the endpoint used to send this request",
       "type": "String"
     }

   },
     "rules": [
      {
       "conditions": [
        {
         "fn": "isSet",
         "argv": [
          {
           "ref": "Endpoint"
          }
         ]
        }
       ],
       "type": "tree",
       "rules": [
        {
         "conditions": [
          {
           "fn": "booleanEquals",
           "argv": [
            {
             "ref": "UseFIPS"
            },

            true
           ]
          }
         ],
         "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
         "type": "error"

```
        },
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ],

          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
          "type": "error"
        },
        {
          "conditions": [],
          "endpoint": {
            "url": {
              "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [

            {
              "ref": "Region"
            }
          ]
        }
      ],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
```

```
                          "argv": [
                            {
                              "ref": "Region"
                            }
                          ],
                          "assign": "PartitionResult"
                        }
                      ],
                      "type": "tree",

          "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                {
                                  "ref": "UseFIPS"
                                },
                                true
                              ]
                            },
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                {
                                  "ref": "UseDualStack"

                                },
                                true
                              ]
                            }
                          ],
                          "type": "tree",
                          "rules": [
                            {
                              "conditions": [
                                {
                                  "fn": "booleanEquals",
                                  "argv": [
                                    true,
                                    {
                                      "fn": "getAttr",

        "argv": [

                                        {
                                          "ref": "PartitionResult"
                                        },
```

```
                                              "supportsFIPS"
                                            ]
                                          }
                                        ]
                                      },
                                      {
                                        "fn": "booleanEquals",
                                        "argv": [
                                          true,
                                          {

              "fn": "getAttr",
                                            "argv": [
                                              {
                                                "ref": "PartitionResult"
                                              },
                                              "supportsDualStack"
                                            ]
                                          }
                                        ]
                                      }
                                    ],
                                    "type": "tree",
                                    "rules": [
                                      {

              "conditions": [],
                                        "endpoint": {
                                          "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                          "properties": {},
                                          "headers": {}
                                        },
                                        "type": "endpoint"
                                      }
                                    ]
                                  },
                                  {
                                    "conditions": [],
                                    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                                     "type": "error"
                                  }
                                ]
                              },
                              {
                                "conditions": [
                                  {
```

```
                                        "fn": "booleanEquals",
                                        "argv": [
                                          {
                                            "ref": "UseFIPS"
                                          },
                                          true
                                        ]
                                      }
                                  ],
                                  "type": "tree",

                          "rules": [
                                      {
                                        "conditions": [
                                          {
                                            "fn": "booleanEquals",
                                            "argv": [
                                              true,
                                              {
                                                "fn": "getAttr",
                                                "argv": [
                                                  {
                                                    "ref": "PartitionResult"
                                                  },
                                                  "supportsFIPS"

                                                ]
                                              }
                                            ]
                                          }
                                        ],
                                        "type": "tree",
                                        "rules": [
                                          {
                                            "conditions": [],
                                            "endpoint": {
                                              "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                              "properties": {},
                                              "headers": {}

                },

                                              "type": "endpoint"
                                          }
                                        ]
                                      },
                                      {
                                        "conditions": [],
```

```
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {

                                "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {

                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
```

```
                        },
                        "type": "endpoint"
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "error": "DualStack is enabled but this partition does
not support DualStack",
                    "type": "error"
                  }
                ]
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ]
          }
        ]
      },
      {
        "conditions":
[],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
      }
    ]
  },
  "smithy.rules#endpointTests": {
    "testCases": [
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": true
```

```
                    }
                },
                {
                    "documentation":
"For region us-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                        }
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                        }
                    },

            "params": {
                    "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                        }
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
                    "expect": {
                        "endpoint": {
```

```
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },

    "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
```

```
              "UseDualStack": false
            }
          },
          {
              "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                }
              },

      "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
          },
          {
              "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
          },
          {
              "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
          },
          {
              "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
```

```
                    "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
                }
            },

    "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                "expect":

{
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                    }
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled but this partition does not support DualStack"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack":

true
                }
            },
            {
```

```
                        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                        "expect": {
                          "endpoint": {
                            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                          }
                        },
                        "params": {
                          "Region": "us-iso-east-1",
                          "UseFIPS": false,
                          "UseDualStack": false
                        }
                      },
                      {
                        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                        "expect": {
                          "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
                        },
                        "params": {
                          "Region": "us-isob-east-1",
                          "UseFIPS": true,
                          "UseDualStack": true
                        }
                      },
                      {
                        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                        "expect": {
                          "endpoint": {
                            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                          }
                        },
                        "params": {
                          "Region": "us-isob-east-1",
                          "UseFIPS": true,
                          "UseDualStack": false
                        }

      },
                      {
                        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
                        "expect": {
                          "error": "DualStack is enabled but this partition does not support DualStack"
                        },
                        "params": {
                          "Region": "us-isob-east-1",
                          "UseFIPS": false,
                          "UseDualStack": true
                        }
```

```
        },
        {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                }
            },
            "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {

            "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
```

```
                    "Region": "us-east-1",
                     "UseFIPS": true,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                  "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
             },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated":
{

            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
        "type": "structure",
```

```
    "members": {
      "IdentityProviderSummaries": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
```

```
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "InstanceSummaries"
            }
          }
        },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
             "traits": {
               "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
             }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
          "type":
"structure",
        "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
            }
          },
```

```
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListProductSubscriptions"
        },
        "smithy.api#paginated": {
```

```
                    "inputToken": "NextToken",
                    "outputToken": "NextToken",
                    "pageSize": "MaxResults",
                    "items": "ProductUserSummaries"
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
            "type": "structure",
            "members": {
                "Product": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
                        "smithy.api#required": {}
                    }
                },
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                        "smithy.api#required": {}
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                    "traits": {

  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n    more sets of key-value pairs that you specify.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
```

```
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
          },

      "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/user/ListUserAssociations"
                },
                "smithy.api#paginated":
{
                    "inputToken": "NextToken",
                    "outputToken": "NextToken",
                    "pageSize": "MaxResults",
                    "items": "InstanceUserSummaries"
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
            "type": "structure",
            "members": {
                "InstanceId": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                        "smithy.api#required": {}
                    }
                },
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                        "smithy.api#required": {}
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify.</p>"
```

```
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
            "type": "structure",
            "members": {
                "InstanceUserSummaries": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                    "traits": {
                        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                    }
                },
                "NextToken": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
            "type": "structure",
            "members": {

                "Username": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Product": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                        "smithy.api#required": {}
                    }
                },
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
```

```
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }


      },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for a user.</p>",
              "smithy.api#required": {}
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
```

        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {

```
                  "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                  "smithy.api#required": {}
                }
              },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n      subnets to provision VPC endpoints.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                  "smithy.api#required": {}
                }
              }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
          }
        },
```

```
        "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
            "type": "string",


      "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n     your VPC and the VPC endpoint for activation servers.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
```

such as the\n   subnets to provision VPC endpoints, and the security group ID that
is associated with the VPC\n   endpoints. The security group should permit inbound TCP port 1688
communication from resources\n   in the VPC.</p>"
                }
            },
            "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
                "type": "operation",
                "input": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
                },
                "output": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
                },
                "errors": [
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
                    },
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
                    }
                ],
                "traits": {
                    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n   <note>\n   <p>Your estimated bill for charges on the number of users and related costs
will take 48\n   hours to appear for billing periods that haven't closed (marked as Pending billing status)
in Amazon Web Services Billing.
For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n   monthly
charges in the <i>Amazon Web Services Billing User Guide</i>.</p>\n   </note>",
                    "smithy.api#http": {
                        "code": 200,
                        "method": "POST",
                        "uri": "/user/StartProductSubscription"

```
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
```

```
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
```

```
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required":
{}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation":
"<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
```

```
      "traits": {
        "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {

          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
```

```
identity\n     provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n     update any combination of settings in a single operation such as the:</p>\n
<ul>\n        <li>\n              <p>Subnets which you want to add to provision VPC endpoints.</p>\n          </li>\n
    <li>\n              <p>Subnets which you want to remove the VPC endpoints from.</p>\n          </li>\n
<li>\n              <p>Security group ID which permits traffic to the VPC endpoints.</p>\n          </li>\n     </ul>",
              "smithy.api#required": {}
            }
          }
        }
      }
    },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        }
```

```
        },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
            "AddSubnets": {

            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n    require connectivity to activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID
that allows inbound TCP port 1688 communication between resources in\n    your VPC and the VPC endpoints for
activation servers.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as\n    the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```

```
{
    "smithy": "2.0",
    "metadata": {
        "suppressions": [
            {
                "id": "HttpMethodSemantics",
                "namespace": "*"
            },
            {
                "id": "HttpResponseCodeSemantics",
                "namespace": "*"
            },
            {
                "id": "PaginatedTrait",
                "namespace": "*"
            },
            {
                "id": "HttpHeaderTrait",
                "namespace": "*"
            },
            {
                "id": "HttpUriConflict",
                "namespace": "*"
            },
            {
                "id": "Service",
                "namespace": "*"
            }
        ]
    },
    "shapes": {
        "com.amazonaws.licensemanager#AWSLicenseManager": {
            "type": "service",
            "version": "2018-08-01",
            "operations": [
                {
                    "target": "com.amazonaws.licensemanager#AcceptGrant"
                },

                {
                    "target": "com.amazonaws.licensemanager#CheckInLicense"
                },
                {
                    "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
                },
                {
                    "target": "com.amazonaws.licensemanager#CheckoutLicense"
                },
```

```
    {
      "target": "com.amazonaws.licensemanager#CreateGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateGrantVersion"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
     },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateToken"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
      "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    }
},
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
    {
      "target": "com.amazonaws.licensemanager#GetGrant"
    }
```

        },
        {
          "target": "com.amazonaws.licensemanager#GetLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
        },
        {
          "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
        },
        {
          "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
        },
        {
          "target": "com.amazonaws.licensemanager#GetLicenseUsage"
        },
        {
          "target": "com.amazonaws.licensemanager#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
        },
        {
          "target": "com.amazonaws.licensemanager#ListDistributedGrants"
        },
        {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
        },
        {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
        },
        {
          "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
        },
        {
          "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
        },
        {
          "target": "com.amazonaws.licensemanager#ListLicenses"
        },
        {
          "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
        },
        {

          "target": "com.amazonaws.licensemanager#ListLicenseVersions"
        },
        {

```
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
    },
    {
      "target": "com.amazonaws.licensemanager#ListResourceInventory"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#ListTokens"
    },
    {
      "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
     },
    {
      "target": "com.amazonaws.licensemanager#RejectGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
    }
],
"traits": {
    "aws.api#service": {
```

          "sdkId": "License Manager",
        "arnNamespace": "license-manager",
        "cloudFormationName": "LicenseManager",
        "cloudTrailEventSource": "licensemanager.amazonaws.com",
        "endpointPrefix": "license-manager"
      },
      "aws.auth#sigv4": {
        "name": "license-manager"
      },
      "aws.protocols#awsJson1_1": {},
      "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n   Amazon Web Services accounts and on-premises servers.</p>",
      "smithy.api#title": "AWS License Manager",
      "smithy.api#xmlNamespace": {
        "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
      },
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
          "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
          "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
            "type": "Boolean"
          },
          "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
          }

```
                },
            "rules": [
              {
                  "conditions": [
                      {
                          "fn": "isSet",
                          "argv": [
                              {
                                  "ref": "Endpoint"
                              }
                          ]

                      }
                  ],
                  "type": "tree",
                  "rules": [
                      {
                          "conditions": [
                              {
                                  "fn": "booleanEquals",
                                  "argv": [
                                      {
                                          "ref": "UseFIPS"
                                      },
                                      true
                                  ]
                              }
                          ],
                          "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                          "type": "error"
                      },
                      {

"conditions": [
                              {
                                  "fn": "booleanEquals",
                                  "argv": [
                                      {
                                          "ref": "UseDualStack"
                                      },
                                      true
                                  ]
                              }
                          ],
                          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                          "type": "error"
                      },
                      {
```

```
                      "conditions": [],
                      "endpoint": {
                        "url": {
                           "ref": "Endpoint"
                         },
                         "properties": {},
                         "headers": {}
                      },
                      "type": "endpoint"
                   }
                ]
             },
             {
                "conditions": [
                   {
                      "fn": "isSet",
                      "argv": [
                         {
                            "ref": "Region"
                         }
                      ]
                   }
                ],
                "type": "tree",
                "rules": [
                   {
                      "conditions": [
                         {
                             "fn": "aws.partition",
                            "argv": [
                               {
                                  "ref": "Region"
                               }
                            ],
                            "assign": "PartitionResult"
                         }
                      ],
                      "type": "tree",
                      "rules": [
                         {
                            "conditions": [
                               {
                                  "fn": "booleanEquals",
                                  "argv": [
                                     {
                                        "ref":
"UseFIPS"
                                     },
```

```
                          true
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              true,
                              {
                                "fn": "getAttr",
                                "argv": [
                                  {
                                    "ref": "PartitionResult"
                                  },
                                  "supportsFIPS"
                                ]

                              }
                            ]
                          },
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              true,
                              {
                                "fn": "getAttr",
                                "argv": [
                                  {
                                    "ref": "PartitionResult"
                                  },

"supportsDualStack"

                                ]
                              }
                            ]
```

```
                            }
                          ],
                          "type": "tree",
                          "rules": [
                            {
                              "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}

                              },
                                "type": "endpoint"
                            }
                          ]
                        },
                        {
                          "conditions": [],
                          "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                          "type": "error"
                        }
                      ]
                    },
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",

                   "argv": [
                            {
                              "ref": "UseFIPS"
                            },
                            true
                          ]
                        }
                      ],
                      "type": "tree",
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
```

```
                              {
                                    "fn": "getAttr",
                                    "argv": [
                                       {
                                          "ref": "PartitionResult"
                                       },
                                       "supportsFIPS"
                                    ]
                              }
                          ]
                     }
                ],
                "type": "tree",
                "rules": [

                    {

                          "conditions": [],
                          "endpoint": {
                             "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                             "properties": {},
                             "headers": {}
                          },
                          "type": "endpoint"
                    }
                ]
             },
             {
                "conditions": [
                   {
                      "fn": "booleanEquals",
                      "argv": [
                         {
                            "ref": "UseDualStack"
                         },
                         true
                      ]
                   }
                ],
                "type": "tree",
```

"error": "FIPS is enabled but this partition does not support FIPS",
"type": "error"

```
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ],
    "type": "tree",
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ]
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
]
},
{
  "conditions": [],
  "endpoint": {
    "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
```

```
                },
                "type": "endpoint"
              }
            ]
          }
        ]
      },
      {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
      }
    ]
  },
  "smithy.rules#endpointTests": {
    "testCases": [
      {
        "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint":

{

              "url": "https://license-manager.af-south-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "af-south-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-northeast-1.amazonaws.com"
```

```
            }
          },
          "params": {
            "Region": "ap-northeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-3.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },

        {
          "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
```

```
        {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-1.amazonaws.com"

}
        },
        "params": {
          "Region": "ap-southeast-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-southeast-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-southeast-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {

"documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ap-southeast-3.amazonaws.com"
          }
        },
        "params": {
          "Region": "ap-southeast-3",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.ca-central-1.amazonaws.com"
          }
        },
```

```
            "params": {
                "Region": "ca-central-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.eu-central-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "eu-central-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region eu-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.eu-north-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "eu-north-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager.eu-south-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "eu-south-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
```

```
{
    "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.eu-west-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "eu-west-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
    "expect": {

        "endpoint": {
            "url": "https://license-manager.eu-west-2.amazonaws.com"
        }
    },
    "params": {
        "Region": "eu-west-2",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.eu-west-3.amazonaws.com"
        }
    },
    "params": {
        "Region": "eu-west-3",
        "UseFIPS": false,

"UseDualStack": false
    }
},
{
    "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.me-south-1.amazonaws.com"
        }
    },
```

```
      "params": {
        "Region": "me-south-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.sa-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "sa-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
```

```
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-east-2.amazonaws.com"
          }
        }
      },
      "params": {
        "Region": "us-east-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-east-2.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-2",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region
us-west-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-west-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-west-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-west-1.amazonaws.com"
        }
      },
      "params": {
```

```
          "Region": "us-west-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
          },
          {
            "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-west-2.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-west-2",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

"expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.us-west-2.amazonaws.com"
                }
            },
            "params": {
              "Region": "us-west-2",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,

              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-northwest-1",
        "UseFIPS": false,
        "UseDualStack": false
      }

}
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
```

          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {

```
              "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-gov-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-west-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-west-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-gov-east-1.api.aws"
            }
          },
      "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
```

```
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-gov-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS": true,
               "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
```

```
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled
but this partition does not support DualStack"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },

            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
```

```
          },

     "params": {
               "Region": "us-isob-east-1",
               "UseFIPS": true,
               "UseDualStack": false
             }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

     "expect": {
               "endpoint": {
                  "url": "https://example.com"
                }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                 "url": "https://example.com"
               }
            },
            "params": {
              "UseFIPS": false,

    "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
```

```
                    },
                    {
                        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                        "expect": {
                            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                        },
                        "params": {
                            "Region": "us-east-1",
                            "UseFIPS": true,
                            "UseDualStack": false,
                            "Endpoint": "https://example.com"
                        }
                    },
                    {
                        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                        "expect": {
                            "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
                        },
                        "params": {
                            "Region": "us-east-1",
                            "UseFIPS": false,
                            "UseDualStack": true,
                            "Endpoint": "https://example.com"
                        }
                    },
                    {
                        "documentation": "Missing region",
                        "expect": {
                            "error": "Invalid Configuration: Missing Region"
                        }
                    }
                ],
                "version": "1.0"
            }
        }
    },
    "com.amazonaws.licensemanager#AcceptGrant": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

},
        "output": {
            "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
        },
        "errors": [
            {
```

```
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
          }
        },
        "com.amazonaws.licensemanager#AcceptGrantRequest": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#AcceptGrantResponse": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Grant
```

```
ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#AccessDeniedException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {

 "code": "ServiceAccessDenied",
          "httpResponseCode": 401
        },
        "smithy.api#documentation": "<p>Access to resource denied.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 401
      }
    },
    "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
      "type": "enum",
      "members": {
        "DISTRIBUTED_GRANTS_ONLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
          }
        },
        "ALL_GRANTS_PERMITTED_BY_ISSUER": {
```

```
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#AllowedOperation": {

    "type": "enum",
        "members": {
          "CREATE_GRANT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CreateGrant"
            }
          },
          "CHECKOUT_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CheckoutLicense"
            }
          },
          "CHECKOUT_BORROW_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CheckoutBorrowLicense"
            }
          },
          "CHECK_IN_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CheckInLicense"
            }
          },
          "EXTEND_CONSUMPTION_LICENSE":
{
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "ExtendConsumptionLicense"
              }
            },
          "LIST_PURCHASED_LICENSES": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ListPurchasedLicenses"
            }
          },
```

```
              "CREATE_TOKEN": {
                "target": "smithy.api#Unit",
                "traits": {
                  "smithy.api#enumValue": "CreateToken"
                }
              }
            }
          }
        },
        "com.amazonaws.licensemanager#AllowedOperationList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#AllowedOperation"
          },
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max":
7
            }
          }
        },
        "com.amazonaws.licensemanager#Arn": {
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 2048
            },
            "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
          }
        },
        "com.amazonaws.licensemanager#ArnList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Arn"
          }
        },
        "com.amazonaws.licensemanager#AuthorizationException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError":
```

```
      {
                "code": "AuthorizationFailure",
                "httpResponseCode": 403
            },
            "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n        policy associated with this account.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 403
        }
    },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
        "type": "structure",
        "members": {
            "LastRunTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes automated discovery.</p>"
        }
    },

  "com.amazonaws.licensemanager#Boolean": {
        "type": "boolean",
        "traits": {
            "smithy.api#default": false
        }
    },
    "com.amazonaws.licensemanager#BorrowConfiguration": {
        "type": "structure",
        "members": {
            "AllowEarlyCheckIn": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxTimeToLiveInMinutes": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
                    "smithy.api#required": {}
                }
            }
```

      },
          "traits": {
            "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
          }
      },
      "com.amazonaws.licensemanager#BoxBoolean": {
        "type": "boolean"
      },
      "com.amazonaws.licensemanager#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanager#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanager#CheckInLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }

```
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
      }
    },
    "com.amazonaws.licensemanager#CheckInLicenseRequest":
{
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckInLicenseResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
      },
      {
        "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
    }
  },
  "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
    "type": "structure",
    "members":
{
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
          "smithy.api#required": {}
        }
```

```
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
            "smithy.api#required": {}
          }
        },
        "DigitalSignatureMethod": {
          "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
          "traits": {
            "smithy.api#documentation": "<p>Digital signature method. The possible
value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
            "smithy.api#required": {}
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "CheckoutMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>Information about constraints.</p>"
          }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
```

```
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
        }
      },
      "LicenseConsumptionToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License consumption token.</p>"
        }
      },
      "EntitlementsAllowed":
{
        "target": "com.amazonaws.licensemanager#EntitlementDataList",
        "traits": {
          "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
        }
      },
      "NodeId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Node ID.</p>"
        }
      },
      "SignedToken": {
        "target": "com.amazonaws.licensemanager#SignedToken",
        "traits": {
          "smithy.api#documentation": "<p>Signed token.</p>"
        }
      },
      "IssuedAt": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
        }
      },
      "Expiration": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
        }
      },
      "CheckoutMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
          "smithy.api#documentation": "<p>Information about constraints.</p>"
        }
      }
    },
```

```
        "traits": {
          "smithy.api#output": { }
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target":
"com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n           <p>If the
account that created the license is the same that is performing the check out, you must\n           specify the account
```

as the beneficiary.</p>\n   </note>"
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
        "type": "structure",
        "members": {
          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product
 SKU.</p>",
              "smithy.api#required": {}
            }
          },
          "CheckoutType": {
            "target": "com.amazonaws.licensemanager#CheckoutType",
            "traits": {
              "smithy.api#documentation": "<p>Checkout type.</p>",
              "smithy.api#required": {}
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>",

      "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"

```
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          }
        }
      },
      "traits": {

    "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
      "type": "structure",
      "members": {
        "CheckoutType": {
          "target": "com.amazonaws.licensemanager#CheckoutType",
          "traits": {
            "smithy.api#documentation": "<p>Checkout type.</p>"
          }
        },
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },
        "EntitlementsAllowed": {
          "target": "com.amazonaws.licensemanager#EntitlementDataList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
          }
        },
         "SignedToken": {
          "target": "com.amazonaws.licensemanager#SignedToken",
          "traits": {
            "smithy.api#documentation": "<p>Signed token.</p>"
          }
        },
        "NodeId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Node ID.</p>"
          }
        },
        "IssuedAt": {
```

```
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
              }
            },
            "Expiration": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
              }
            },
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#CheckoutType": {
          "type": "enum",
          "members": {
            "PROVISIONAL": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "PROVISIONAL"
              }
            },
            "PERPETUAL": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "PERPETUAL"
              }
            }
          }
        },
        "com.amazonaws.licensemanager#ClientRequestToken": {
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 36
            }
          }
```

```
      },
      "com.amazonaws.licensemanager#ClientToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 2048
          },
          "smithy.api#pattern": "^\\S+$"
        }
      },
      "com.amazonaws.licensemanager#ConflictException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "ConflictException",
            "httpResponseCode":
409
          },
          "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 409
        }
      },
      "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
        "type": "structure",
        "members": {
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
            }
          },
          "ConsumedLicenses": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
            }
          }
        },
```

```
"traits": {
        "smithy.api#documentation": "<p>Details about license consumption.</p>"
    }
},
"com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
    }
},
"com.amazonaws.licensemanager#ConsumptionConfiguration": {
    "type": "structure",
    "members": {
        "RenewType": {
            "target": "com.amazonaws.licensemanager#RenewType",
            "traits": {
                "smithy.api#documentation": "<p>Renewal frequency.</p>"
            }
        },
        "ProvisionalConfiguration": {
            "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
            }
        },
        "BorrowConfiguration": {
            "target": "com.amazonaws.licensemanager#BorrowConfiguration",
            "traits": {
                "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
    }
},
"com.amazonaws.licensemanager#CreateGrant": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
```

```
            {
                "target":
  "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
  license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
  organizational unit (OU). For more
   information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
  licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateGrantRequest": {
        "type": "structure",
        "members": {
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
  idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant name.</p>",
                    "smithy.api#required": {}

    }
        },
        "LicenseArn": {
```

```
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Principals": {
          "target": "com.amazonaws.licensemanager#PrincipalArnList",
          "traits": {
            "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n         (ARN):</p>\n      <ul>\n         <li>\n            <p>An Amazon Web Services
account, which includes only the account specified.</p>\n         </li>\n      </ul>\n      <ul>\n         <li>\n
   <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n         </li>\n      </ul>\n
<ul>\n
      <li>\n            <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n      </ul>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },

    "com.amazonaws.licensemanager#CreateGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
```

```
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateGrantVersion":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
```

```
          ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
      },
    "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
        "type": "structure",
        "members": {
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
            }
          },
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          },
          "GrantName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant name.</p>"
            }
          },
          "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
              "smithy.api#documentation": "<p>Grant status reason.</p>"
```

```
                }
            },
            "SourceVersion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Current version of the grant.</p>"
                }
            },
            "Options": {
                "target": "com.amazonaws.licensemanager#Options",
                "traits": {
                    "smithy.api#documentation": "<p>The options specified for the grant.</p>"
                }
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#CreateGrantVersionResponse": {
    "type": "structure",
    "members": {
        "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
                "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
            }
        },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>New version of the grant.</p>"
            }
        }
    },
    "traits":
{
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#CreateLicense": {
    "type": "operation",
```

```
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
 all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of
licenses purchased and used.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
```

          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used. Exceeding a hard limit\n   blocks the launch of new instances.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example, #allowedTenancy=EC2-DedicatedHost). The available rules \n   vary by dimension, as follows.</p>\n   \n   \n   <p>\n   <code>Cores</code> dimension: <code>allowedTenancy</code> |\n   <code>licenseAffinityToHost</code> |\n   <code>maximumCores</code> | <code>minimumCores</code>\n   </p>\n   \n   \n   <p>\n   <code>Instances</code> dimension: <code>allowedTenancy</code> | \n   <code>maximumCores</code> | <code>minimumCores</code> |\n   <code>maximumSockets</code> | <code>minimumSockets</code> |\n   <code>maximumVcpus</code> | <code>minimumVcpus</code>\n   </p>\n   \n   \n   <p>\n   <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n   <code>licenseAffinityToHost</code> |\n   <code>maximumSockets</code> | <code>minimumSockets</code>\n   </p>\n   \n   \n   <p>\n   <code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n   <code>honorVcpuOptimization</code> | \n   <code>maximumVcpus</code> | <code>minimumVcpus</code>\n   </p>\n   \n   \n   <p>The unit for <code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for <code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n   <code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n   <code>True</code> and <code>False</code>.</p>"
          }
        },

    "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is uninstalled.</p>"

```
          }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>Product information.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {

          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
            "smithy.api#required": {}
          }
        },
        "SourceLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n     For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
            "smithy.api#required": {}
          }
        },
        "DestinationLicenseContext":
{

          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
```

```
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
            "type": "structure",
            "members": {
                "LicenseConversionTaskId": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

                "traits": {
                        "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
```

```
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a report generator.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "ReportGeneratorName":
  {
              "target": "com.amazonaws.licensemanager#ReportGeneratorName",
              "traits": {
                "smithy.api#documentation": "<p>Name of the report generator.</p>",
                "smithy.api#required": {}
              }
          },
          "Type": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
              "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n      <ul>\n         <li>\n            <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n         </li>\n         <li>\n            <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n         </li>\n
</ul>",
              "smithy.api#required": {}

  }
          },
          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits": {
              "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientRequestToken",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Unique,
case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
```

```json
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product
name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>",
            "smithy.api#required": {}
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#Issuer",
          "traits": {
            "smithy.api#documentation": "<p>License issuer.</p>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region for the license.</p>",
            "smithy.api#required": {}

  }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
```

```
              "traits": {
                "smithy.api#documentation": "<p>License beneficiary.</p>",
                "smithy.api#required": {}
              }
            },
          "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n          running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n          usage.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license.</p>"
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
```

```
      "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
          }
      }
  },
  "traits": {
      "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#CreateLicenseVersion": {
  "type": "operation",
  "input": {
      "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
  },
  "output": {
      "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
  },
  "errors": [
      {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
          "target": "com.amazonaws.licensemanager#ConflictException"
      },
      {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
          "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
          "target": "com.amazonaws.licensemanager#ValidationException"
      }
  ],
  "traits": {
      "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
  }
```

```json
        },
      "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target":
"com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product name.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits":
{
              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
              "smithy.api#required": {}
            }
```

```json
            },
            "LicenseMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license.</p>"
                }
            },
            "Entitlements": {
                "target": "com.amazonaws.licensemanager#EntitlementList",
                "traits": {
                    "smithy.api#documentation": "<p>License entitlements.</p>",
                    "smithy.api#required": {}
                }
            },
            "ConsumptionConfiguration": {
                "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for workloads with offline\n        usage.</p>",
                    "smithy.api#required": {}
                }
            },

    "Status": {
                "target": "com.amazonaws.licensemanager#LicenseStatus",
                "traits": {
                    "smithy.api#documentation": "<p>License status.</p>",
                    "smithy.api#required": {}
                }
            },
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            },
            "SourceVersion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Current version of the license.</p>"
                }
            }
        },
        "traits": {
```

"smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New version of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
{}
        }
      },
      "com.amazonaws.licensemanager#CreateToken": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },

```
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },

      {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateTokenRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits":
 {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
              "smithy.api#required": {}
            }
          },
          "RoleArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n        License Manager does not check whether the roles are in use.</p>"
            }
          },
          "ExpirationInDays": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
              "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
            }
```

```
        },
        "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
            "traits": {
              "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n        to the amr claim of the JWT token.</p>"
            }
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateTokenResponse": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",

          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
        "TokenType": {
          "target": "com.amazonaws.licensemanager#TokenType",
          "traits": {
            "smithy.api#documentation": "<p>Token type.</p>"
          }
        },
        "Token": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DateTime": {
```

```json
        "type": "timestamp"
    },
    "com.amazonaws.licensemanager#DatetimeRange": {
        "type": "structure",
        "members": {
            "Begin": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Start of the time range.</p>",
                    "smithy.api#required": {}
                }
            },
            "End": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>End of the time range.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
        }
    },
    "com.amazonaws.licensemanager#DeleteGrant": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
```

```
              "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
              "target": "com.amazonaws.licensemanager#ValidationException"
          }
      ],
      "traits": {
          "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
      }
  },
  "com.amazonaws.licensemanager#DeleteGrantRequest": {
      "type": "structure",
      "members": {
          "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                  "smithy.api#required": {}
              }
          },
          "StatusReason": {
              "target": "com.amazonaws.licensemanager#StatusReasonMessage",
              "traits": {
                  "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
              }
          },
          "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                  "smithy.api#documentation": "<p>Current version of the
grant.</p>",
                  "smithy.api#required": {}
              }
          }
      },
      "traits": {
          "smithy.api#input": {}
      }
  },
  "com.amazonaws.licensemanager#DeleteGrantResponse": {
      "type": "structure",
      "members": {
          "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                  "smithy.api#documentation": "<p>Grant ARN.</p>"
              }
```

```
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation":
"<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
        },

      "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
        "type": "structure",
        "members":
{
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
              "smithy.api#documentation": "<p>ID of the license configuration.</p>",
              "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
```

```
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n    <p>This action
deletes the report generator, which stops it from generating future reports.\n    The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
            "type": "structure",
            "members": {
                "LicenseManagerReportGeneratorArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {

    "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                        "smithy.api#required": {}
                    }
                },
                "SourceVersion": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Current version of the license.</p>",
                        "smithy.api#required":
```

```
{ }
            }
          }
        },
        "traits": {
          "smithy.api#input": { }
        }
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseResponse": {
      "type": "structure",
      "members": {
        "Status": {
          "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "DeletionDate": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": { }
      }
    },
    "com.amazonaws.licensemanager#DeleteToken": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#RedirectException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },

    {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenRequest": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteTokenResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}

    }
    },
    "com.amazonaws.licensemanager#DigitalSignatureMethod": {
      "type": "enum",
      "members": {
        "JWT_PS384": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "JWT_PS384"
          }
```

```
            }
          }
        },
        "com.amazonaws.licensemanager#Entitlement": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Entitlement name.</p>",
                "smithy.api#required": {}
              }
            },
            "Value": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
              }
            },
            "MaxCount": {
              "target": "com.amazonaws.licensemanager#Long",
              "traits": {
                "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
              }
            },
            "Overage": {
              "target": "com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
              }
            },
            "Unit": {
              "target": "com.amazonaws.licensemanager#EntitlementUnit",
              "traits": {
                "smithy.api#documentation": "<p>Entitlement unit.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowCheckIn": {
              "target":
"com.amazonaws.licensemanager#BoxBoolean",
              "traits": {
                "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
              }
            }
          },
          "traits": {
```

```
              "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
          }
      },
      "com.amazonaws.licensemanager#EntitlementData": {
          "type": "structure",
          "members": {
              "Name": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                      "smithy.api#documentation": "<p>Entitlement data name.</p>",
                      "smithy.api#required": {}
                  }
              },
              "Value": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                      "smithy.api#documentation": "<p>Entitlement
data value.</p>"
                  }
              },
              "Unit": {
                  "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
                  "traits": {
                      "smithy.api#documentation": "<p>Entitlement data unit.</p>",
                      "smithy.api#required": {}
                  }
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
          }
      },
      "com.amazonaws.licensemanager#EntitlementDataList": {
          "type": "list",
          "member": {
              "target": "com.amazonaws.licensemanager#EntitlementData"
          }
      },
      "com.amazonaws.licensemanager#EntitlementDataUnit": {
          "type": "enum",
          "members": {
              "COUNT": {
                  "target": "smithy.api#Unit",
                  "traits": {

"smithy.api#enumValue": "Count"
                  }
              },
```

```
"NONE": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "None"
  }
},
"SECONDS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Seconds"
  }
},
"MICROSECONDS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Microseconds"
  }
},
"MILLISECONDS": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Milliseconds"
  }
},
"BYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Bytes"
  }
},
"KILOBYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Kilobytes"
  }
},
"MEGABYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Megabytes"
  }
},
"GIGABYTES": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Gigabytes"
  }
},
```

```
      "TERABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabytes"
        }
      },
      "BITS":
{

        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits"
        }
      },
      "KILOBITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits"
        }
      },
      "MEGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabits"
        }
      },
      "GIGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits"
        }
      },
      "TERABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits"

}
      },
      "PERCENT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Percent"
        }
      },
      "BYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes/Second"
```

```
          }
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
           "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabytes/Second"
          }
        },
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
```

```
              "smithy.api#enumValue": "Gigabits/Second"
            }
          },
          "TERABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Terabits/Second"
            }
          },
          "COUNT_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count/Second"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#EntitlementList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Entitlement"
        }
      },
      "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#EntitlementUnit": {
        "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
            "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue": "None"
      }
    },
    "SECONDS":
{
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Seconds"
      }
    },
    "MICROSECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Microseconds"
      }
    },
    "MILLISECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Milliseconds"
      }
    },
    "BYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
```

```
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabytes"
        }
      },
      "BITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits"
        }
      },
      "KILOBITS": {
        "target": "smithy.api#Unit",
        "traits": {

          "smithy.api#enumValue": "Kilobits"
        }
      },
      "MEGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabits"
        }
      },
      "GIGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits"
        }
      },
      "TERABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits"
        }
      },
      "PERCENT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Percent"
        }
      },
      "BYTES_PER_SECOND": {
        "target":
"smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes/Second"
        }
```

```
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {

"smithy.api#enumValue": "Terabytes/Second"
          }
        },
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
```

```
            }
        },

    "TERABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Terabits/Second"
            }
        },
        "COUNT_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Count/Second"
            }
        }
    }
},
"com.amazonaws.licensemanager#EntitlementUsage": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Entitlement usage name.</p>",
                "smithy.api#required": {}
            }
        },
        "ConsumedValue": {
            "target": "com.amazonaws.licensemanager#String",

        "traits": {
                "smithy.api#documentation": "<p>Resource usage consumed.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxCount": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
            }
        },
        "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
            "traits": {
                "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
                "smithy.api#required": {}
            }
        }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#EntitlementUsageList":
{
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#EntitlementUsage"
        }
      },
      "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
           },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
        }
      },
      "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
```

```
        "type": "structure",
        "members": {
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>",
              "smithy.api#required": {}
            }

  }
        },
        "DryRun": {
          "target": "com.amazonaws.licensemanager#Boolean",
          "traits": {
            "smithy.api#default": false,
            "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },

        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#FailedDependencyException": {
      "type": "structure",
      "members": {
        "Message": {
```

```
            "target": "com.amazonaws.licensemanager#Message"
          },
          "ErrorCode": {
            "target": "com.amazonaws.licensemanager#String"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "FailedDependency",
            "httpResponseCode": 424
          },
          "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 424
        }
      },
      "com.amazonaws.licensemanager#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#FilterName",
            "traits": {
              "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
            }
          },
          "Values": {
            "target": "com.amazonaws.licensemanager#FilterValues",
            "traits": {
              "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n      describe operation. Filters can be used
to match a set of resources by specific criteria,\n      such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanager#FilterLimitExceededException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
```

```
            "aws.protocols#awsQueryError": {
               "code": "FilterLimitExceeded",
               "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
         }
      },
      "com.amazonaws.licensemanager#FilterList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanager#Filter"
         }
      },
      "com.amazonaws.licensemanager#FilterName": {
         "type": "string"
      },
      "com.amazonaws.licensemanager#FilterValue": {
         "type": "string"
      },
      "com.amazonaws.licensemanager#FilterValues": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanager#FilterValue",
            "traits": {
               "smithy.api#xmlName": "item"
            }
         }
      },
      "com.amazonaws.licensemanager#Filters": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanager#Filter",
            "traits": {
               "smithy.api#xmlName": "item"
            }
         }
      },
      "com.amazonaws.licensemanager#GetAccessToken": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
         },
         "errors": [
```

```
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
  tokens\n        are valid for one hour.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenRequest": {
          "type": "structure",
          "members": {
            "Token": {
              "target": "com.amazonaws.licensemanager#TokenString",
              "traits": {
                "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
                "smithy.api#required": {}
              }
            },
            "TokenProperties": {
              "target": "com.amazonaws.licensemanager#MaxSize3StringList",
              "traits": {
                "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
          "type": "structure",
          "members": {
```

```
          "AccessToken": {
            "target": "com.amazonaws.licensemanager#TokenString",
            "traits": {
              "smithy.api#documentation": "<p>Temporary access token.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetGrant": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetGrantRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetGrantResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetGrantRequest": {
```

```
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrantResponse": {
      "type": "structure",
      "members": {
        "Grant": {
          "target": "com.amazonaws.licensemanager#Grant",
          "traits": {
            "smithy.api#documentation": "<p>Grant details.</p>"
          }
        }
      },
      "traits": {

"smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
```

```
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#GetLicenseConfiguration": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
```

```json
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits":
{
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
        "type": "structure",
        "members": {
          "LicenseConfigurationId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
            }
          },
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
            }
          },
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the license configuration.</p>"
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the license configuration.</p>"
            }
          },
          "LicenseCountingType": {
            "target": "com.amazonaws.licensemanager#LicenseCountingType",
```

```
      "traits": {
        "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
      }
    },
    "LicenseRules": {
      "target": "com.amazonaws.licensemanager#StringList",
      "traits": {
        "smithy.api#documentation": "<p>License rules.</p>"
      }
    },
    "LicenseCount": {
      "target": "com.amazonaws.licensemanager#BoxLong",


      "traits": {
        "smithy.api#documentation": "<p>Number of available licenses.</p>"
      }
    },
    "LicenseCountHardLimit": {
      "target": "com.amazonaws.licensemanager#BoxBoolean",
      "traits": {
        "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
      }
    },
    "ConsumedLicenses": {
      "target": "com.amazonaws.licensemanager#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>License configuration status.</p>"
      }

    },
    "OwnerAccountId": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
      }
    },
    "ConsumedLicenseSummaryList": {
      "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
      }
    },
```

```
"ManagedResourceSummaryList": {
  "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
  "traits": {
    "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
  }
},
"Tags": {
  "target": "com.amazonaws.licensemanager#TagList",
  "traits": {
    "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
  }
},
"ProductInformationList": {
  "target": "com.amazonaws.licensemanager#ProductInformationList",
  "traits": {
    "smithy.api#documentation": "<p>Product information.</p>"
  }
},
"AutomatedDiscoveryInformation": {
  "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
  "traits": {
    "smithy.api#documentation": "<p>Automated discovery information.</p>"
  }
},
"DisassociateWhenNotFound": {
  "target": "com.amazonaws.licensemanager#BoxBoolean",
  "traits": {
    "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
  }
}
}
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanager#GetLicenseConversionTask": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
```

```
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {

        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
          "type": "structure",
          "members": {
            "LicenseConversionTaskId": {
              "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
              "traits": {
                "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
          "type": "structure",
          "members": {
            "LicenseConversionTaskId": {
              "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
              "traits": {
                "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
              }
            },
            "ResourceArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
```

```
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
            }
          },

    "StartTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
            }
          },
          "LicenseConversionTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
            }
          },
          "EndTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
            }
          }
        },
```

```
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
```

```
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
              "smithy.api#required":
{}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {
          "ReportGenerator": {
            "target": "com.amazonaws.licensemanager#ReportGenerator",
            "traits": {
              "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
          }
        },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

```
      },
    "com.amazonaws.licensemanager#GetLicenseResponse": {
      "type": "structure",
      "members": {
        "License": {
          "target": "com.amazonaws.licensemanager#License",
          "traits": {
            "smithy.api#documentation": "<p>License details.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },

"com.amazonaws.licensemanager#GetLicenseUsage": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {

"target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
    }
```

```
          },
      "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
        "type": "structure",
        "members": {
          "LicenseUsage": {
            "target":
"com.amazonaws.licensemanager#LicenseUsage",
            "traits": {
              "smithy.api#documentation": "<p>License usage details.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

          },
          {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "S3BucketArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n   and so on.</p>"

 }
          },
          "SnsTopicArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
            }
          },
          "OrganizationConfiguration": {
            "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n   cross-account discovery.</p>"
            }
          },
          "EnableCrossAccountsDiscovery": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
            }

            },
            "LicenseManagerResourceShareArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n   provides member accounts with access to this share.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#Grant": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantName": {

                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant name.</p>",
                    "smithy.api#required": {}
                }
            },
            "ParentArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Parent ARN.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>License ARN.</p>",
                    "smithy.api#required": {}
                }
            },
            "GranteePrincipalArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {

"smithy.api#documentation": "<p>The grantee principal ARN.</p>",
                "smithy.api#required": {}
            }
        },
        "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                "smithy.api#required": {}
            }
        },
        "GrantStatus": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>",
                "smithy.api#required": {}
            }
        },
        "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
                "smithy.api#documentation": "<p>Grant status reason.</p>"
            }
        },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>",
                "smithy.api#required": {}
            }
        },
        "GrantedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
                "smithy.api#documentation": "<p>Granted operations.</p>",
                "smithy.api#required": {}
            }
        },
        "Options": {
            "target": "com.amazonaws.licensemanager#Options",
            "traits": {
                "smithy.api#documentation": "<p>The options specified for the grant.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation":

"<p>Describes a grant.</p>"
        }
    },
    "com.amazonaws.licensemanager#GrantList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#Grant"
        }
    },
    "com.amazonaws.licensemanager#GrantStatus": {
        "type": "enum",
        "members": {
            "PENDING_WORKFLOW": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "PENDING_WORKFLOW"
                }
            },
            "PENDING_ACCEPT": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "PENDING_ACCEPT"
                }
            },
            "REJECTED": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "REJECTED"

                }
            },
            "ACTIVE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "ACTIVE"
                }
            },
            "FAILED_WORKFLOW": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "FAILED_WORKFLOW"
                }
            },
            "DELETED": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DELETED"
                }

```
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits":
{
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>"
          }
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
```

```json
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#IssuerDetails",
        "traits": {
          "smithy.api#documentation": "<p>Granted license issuer.</p>"
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
        "traits": {
          "smithy.api#documentation": "<p>Granted license status.</p>"
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in ISO8601-UTC format.</p>"
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>"
        }
      },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
        }
      },
      "LicenseMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
```

```
                  "traits": {
                    "smithy.api#documentation": "<p>Granted license metadata.</p>"
                  }
                },
                "CreateTime": {
                  "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                  "traits": {
                    "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

  }
                },
                "Version": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Version of the granted license.</p>"
                  }
                },
                "ReceivedMetadata": {
                  "target": "com.amazonaws.licensemanager#ReceivedMetadata",
                  "traits": {
                    "smithy.api#documentation": "<p>Granted license received metadata.</p>"
                  }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
            }
        },
        "com.amazonaws.licensemanager#GrantedLicenseList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#GrantedLicense"
            }
        },
        "com.amazonaws.licensemanager#ISO8601DateTime":
{
            "type": "string",
            "traits": {
                "smithy.api#length": {
                    "min": 0,
                    "max": 50
                },
                "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
            }
        },
        "com.amazonaws.licensemanager#Integer": {
            "type": "integer"
```

```
        },
      "com.amazonaws.licensemanager#InvalidParameterValueException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidParameterValueProvided",
            "httpResponseCode": 400
          },
          "smithy.api#documentation":
"<p>One or more parameter values are not valid.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#InvalidResourceStateException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidResourceState",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n        <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },

      "com.amazonaws.licensemanager#InventoryFilter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the filter.</p>",
              "smithy.api#required": {}
            }
```

```
        },
        "Condition": {
          "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
          "traits": {
            "smithy.api#documentation": "<p>Condition of the filter.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }

    },
      "traits": {
        "smithy.api#documentation": "<p>An inventory filter.</p>"
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterCondition": {
      "type": "enum",
      "members": {
        "EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EQUALS"
          }
        },
        "NOT_EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "NOT_EQUALS"
          }
        },
        "BEGINS_WITH": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "BEGINS_WITH"
          }
        },
        "CONTAINS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CONTAINS"
          }
        }
      }
```

```
    },
    "com.amazonaws.licensemanager#InventoryFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#InventoryFilter"
      }
    },
    "com.amazonaws.licensemanager#Issuer": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Issuer name.</p>",
            "smithy.api#required": {}
          }
        },
        "SignKey": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
 of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#IssuerDetails": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Issuer name.</p>"
          }
        },
        "SignKey": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
          }
        },
        "KeyFingerprint": {
```

```
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
  }
},
"com.amazonaws.licensemanager#License": {
  "type": "structure",
  "members": {
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
      }
    },
    "LicenseName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {

      "smithy.api#documentation": "<p>License name.</p>"
        }
    },
    "ProductName": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product name.</p>"
      }
    },
    "ProductSKU": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Product SKU.</p>"
      }
    },
    "Issuer": {
      "target": "com.amazonaws.licensemanager#IssuerDetails",
      "traits": {
        "smithy.api#documentation": "<p>License issuer.</p>"
      }
    },
    "HomeRegion": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
```

```
              "smithy.api#documentation": "<p>Home Region of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          },
          "Entitlements": {

    "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>"
            }
          },
          "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>License metadata.</p>"
            }
          },
          "CreateTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>License
creation time.</p>"
            }
          },
```

```
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfiguration": {
      "type": "structure",
      "members": {
        "LicenseConfigurationId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
          }
        },
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
          }
        },
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
          }
        },
        "LicenseRules": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>License rules.</p>"
```

```
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the license configuration.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

          }
        },
        "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
          }
        },
        "ManagedResourceSummaryList": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
```

```
        "traits": {
          "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
        }
      },
      "ProductInformationList": {
        "target": "com.amazonaws.licensemanager#ProductInformationList",
        "traits": {
          "smithy.api#documentation": "<p>Product information.</p>"
        }
      },
      "AutomatedDiscoveryInformation": {

        "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
        "traits": {
          "smithy.api#documentation": "<p>Automated discovery information.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n        consumed and enforced by License Manager. Components include specifications for
the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n        must be associated with a
host), and the number of licenses purchased and used.</p>"
    }
  },
  "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
    "type": "structure",
    "members": {
      "ResourceArn":
{
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Type of server resource.</p>"
        }
      },
      "ResourceOwnerId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
        }
```

```
        },
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {

    "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
          }
        },
        "AmiAssociationScope": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
      "type": "enum",
      "members":
{
        "AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISABLED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
      "type": "structure",
      "members": {
        "ResourceArn": {
```

```
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits":
{
          "smithy.api#documentation": "<p>Type of resource.</p>"
        }
      },
      "ResourceStatus": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Status of the resource.</p>"
        }
      },
      "ResourceOwnerId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
        }
      },
      "AssociationTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
        }
      },

  "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
```

```
            }
        },
    "com.amazonaws.licensemanager#LicenseConfigurations": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#LicenseConfiguration"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionContext": {

    "type": "structure",
        "members": {
            "UsageOperation": {
                "target": "com.amazonaws.licensemanager#UsageOperation",
                "traits": {
                    "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n        </p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {

    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
                }
            },
            "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
                }
            },
            "SourceLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
```

```
            }
          },
          "DestinationLicenseContext": {
            "target":
 "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the conversion task.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "StartTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation":
 "<p>The time the conversion task was started at.</p>"
            }
          },
          "LicenseConversionTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
            }
          },
          "EndTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTaskId": {
```

```
            "type": "string",
            "traits": {

        "smithy.api#length": {
                "min": 0,
                "max": 50
            },
              "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
            }
        },
      "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
          "type": "enum",
          "members": {
            "IN_PROGRESS": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "IN_PROGRESS"
              }
            },
            "SUCCEEDED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "SUCCEEDED"
              }
            },
            "FAILED": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "FAILED"
              }
            }
          }
        },
      "com.amazonaws.licensemanager#LicenseConversionTasks":
{
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTask"
          }
        },
      "com.amazonaws.licensemanager#LicenseCountingType": {
          "type": "enum",
          "members": {
            "VCPU": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "vCPU"
              }
```

```
          },
          "INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Instance"
            }
          },
          "CORE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Core"
            }
          },
          "SOCKET": {
            "target": "smithy.api#Unit",
             "traits": {
              "smithy.api#enumValue": "Socket"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseDeletionStatus": {
        "type": "enum",
        "members": {
          "PENDING_DELETE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_DELETE"
            }
          },
          "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DELETED"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#LicenseList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#License"
        }
      },
      "com.amazonaws.licensemanager#LicenseOperationFailure": {

        "type": "structure",
        "members": {
```

```
"ResourceArn": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
  }
},
"ResourceType": {
  "target": "com.amazonaws.licensemanager#ResourceType",
  "traits": {
    "smithy.api#documentation": "<p>Resource type.</p>"
  }
},
"ErrorMessage": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>Error message.</p>"
  }
},
"FailureTime": {
  "target": "com.amazonaws.licensemanager#DateTime",
  "traits": {

    "smithy.api#documentation": "<p>Failure time.</p>"
  }
},
"OperationName": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>Name of the operation.</p>"
  }
},
"ResourceOwnerId": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
  }
},
"OperationRequestedBy": {
  "target": "com.amazonaws.licensemanager#String",
  "traits": {
    "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
  }
},
"MetadataList": {

  "target": "com.amazonaws.licensemanager#MetadataList",
  "traits": {
```

```
              "smithy.api#documentation": "<p>Reserved.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseOperationFailureList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecification": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

              "smithy.api#required": {}
            }
          },
          "AmiAssociationScope": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecifications": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseSpecification"
        }
      },
      "com.amazonaws.licensemanager#LicenseStatus": {
        "type": "enum",
        "members": {
          "AVAILABLE": {
```

```
              "target": "smithy.api#Unit",

          "traits": {
              "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "PENDING_AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_AVAILABLE"
          }
        },
        "DEACTIVATED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DEACTIVATED"
          }
        },
        "SUSPENDED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUSPENDED"
          }
        },
        "EXPIRED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EXPIRED"
          }
        },

      "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseUsage": {
      "type": "structure",
      "members": {
```

```
      "EntitlementUsages": {
        "target": "com.amazonaws.licensemanager#EntitlementUsageList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlement usages.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

    }
  },
  "com.amazonaws.licensemanager#LicenseUsageException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {
        "code": "LicenseUsageFailure",
        "httpResponseCode": 412
      },
      "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 412
    }
  },
  "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
    },
    "output": {

"target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
      "type": "structure",
```

```
        "members": {
          "LicenseConfigurationAssociations": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
            "traits": {
              "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
            }
          },
          "NextToken": {

 "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListDistributedGrants": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
```

```
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
      "type": "structure",
      "members": {
        "GrantArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {

          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n        <p>\n              <code>LicenseArn</code>\n        </p>\n
    </li>\n        <li>\n        <p>\n              <code>GrantStatus</code>\n        </p>\n        </li>\n
  <li>\n        <p>\n              <code>GranteePrincipalARN</code>\n        </p>\n        </li>\n
<li>\n        <p>\n              <code>ProductSKU</code>\n        </p>\n        </li>\n        <li>\n
<p>\n              <code>LicenseIssuerName</code>\n
        </p>\n        </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
      "type": "structure",
```

```
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",

    "traits": {
              "smithy.api#documentation": "<p>Distributed grant details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
      },
      "errors": [
        {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
      }
```

```
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
           "traits": {
             "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
             "smithy.api#required": {}
           }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
      "type": "structure",
      "members": {
        "LicenseOperationFailureList": {
          "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
          "traits": {
            "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
```

```
        },
    "com.amazonaws.licensemanager#ListLicenseConfigurations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
        },

    "output": {
            "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the license configurations
for your account.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArns": {
                "target": "com.amazonaws.licensemanager#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
```

```
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next
 set of results.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n            <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n            Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n            Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n            </li>\n        <li>\n
<p>\n            <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n        Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
    </li>\n        <li>\n            <p>\n            <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
    "type": "structure",
    "members": {
        "LicenseConfigurations": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurations",
            "traits": {
                "smithy.api#documentation": "<p>Information about the license configurations.</p>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",

         "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
      "type": "structure",
      "members": {
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {

"smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
```

```
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>\n       Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n     </p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTasks": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target":
```

"com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n   \n   \n   <p>\n   <code>LicenseConfigurationArn</code>\n
</p>\n   \n   "
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits":
{
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

```
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          }
        },
        "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
          "type": "structure",
          "members": {
            "ReportGenerators": {
              "target": "com.amazonaws.licensemanager#ReportGeneratorList",
              "traits": {
                "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }


            }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
              },
              {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
```

```
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
          "type": "structure",
          "members": {
            "LicenseSpecifications": {
              "target": "com.amazonaws.licensemanager#LicenseSpecifications",
              "traits": {
                "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
              }
            },
```

```
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#ListLicenseVersions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
  }
},
"com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
  "type": "structure",
  "members": {
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
        "smithy.api#required": {}
```

```
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
             "traits": {
               "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#LicenseList",
            "traits": {
              "smithy.api#documentation": "<p>License details.</p>"
            }
          },

"NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenses": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicensesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicensesResponse"
        },
```

```
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
       },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListLicensesRequest": {
    "type": "structure",
    "members": {
      "LicenseArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
        }
      },

   "Filters": {
        "target": "com.amazonaws.licensemanager#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n      <ul>\n        <li>\n          <p>\n            <code>Beneficiary</code>\n          </p>\n
      </li>\n        <li>\n          <p>\n            <code>ProductSKU</code>\n          </p>\n        </li>\n
   <li>\n          <p>\n            <code>Fingerprint</code>\n          </p>\n        </li>\n        <li>\n
<p>\n            <code>Status</code>\n          </p>\n        </li>\n      </ul>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token
```

for the next set of results.</p>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                  "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicensesResponse": {
          "type": "structure",
          "members": {
            "Licenses": {
              "target": "com.amazonaws.licensemanager#LicenseList",
              "traits": {
                "smithy.api#documentation": "<p>License details.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {

    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListReceivedGrants": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {

```
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
       },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n        recipient as this Amazon Web Services account, your organization, or an organizational unit\n
  (OU) to which this member account belongs.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
     },
    "output": {
      "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
```

```
                    {
                        "target": "com.amazonaws.licensemanager#ServerInternalException"
                    },
                    {
                        "target":
"com.amazonaws.licensemanager#ValidationException"
                    }
                ],
                "traits": {
                    "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
                }
            },
            "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
                "type": "structure",
                "members": {
                    "LicenseArn": {
                        "target": "com.amazonaws.licensemanager#Arn",
                        "traits": {
                            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
                            "smithy.api#required": {}
                        }
                    },
                    "Filters": {
                        "target": "com.amazonaws.licensemanager#FilterList",
                        "traits": {
                            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
    <ul>\n          <li>\n             <p>\n                <code>ParentArn</code>\n             </p>\n          </li>\n
<li>\n             <p>\n                <code>GranteePrincipalArn</code>\n             </p>\n          </li>\n      </ul>"
                        }
                    },
                    "NextToken": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                        }
                    },
                    "MaxResults": {
                        "target": "com.amazonaws.licensemanager#MaxSize100",
                        "traits": {
                            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                        }
                    }
                },
                "traits": {
                    "smithy.api#input": {}
                }
            },
```

"com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
    "type": "structure",
    "members": {
      "Grants": {
        "target": "com.amazonaws.licensemanager#GrantList",
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
    "type": "structure",
    "members": {
      "GrantArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:</p>\n   \n   \n   <p>\n   <code>ProductSKU</code>\n   </p>\n   \n   \n   <p>\n   <code>LicenseIssuerName</code>\n   </p>\n   \n   \n   <p>\n   <code>LicenseArn</code>\n   </p>\n   \n   \n   <p>\n   <code>GrantStatus</code>\n   </p>\n   \n   \n   <p>\n   <code>GranterAccountId</code>\n   </p>\n   \n   "
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },

```
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
      "type": "structure",
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",
          "traits": {
            "smithy.api#documentation": "<p>Received grant details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicenses": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists received licenses.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {

"target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
```

```
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>Beneficiary</code>\n            </p>\n
</li>\n          <li>\n            <p>\n              <code>ProductSKU</code>\n
</p>\n          </li>\n        </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#GrantedLicenseList",
            "traits": {
              "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
        "type": "structure",
        "members": {
            "LicenseArns": {
                "target": "com.amazonaws.licensemanager#ArnList",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
                }
            },

            "Filters": {
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n            <li>\n            <p>\n                <code>ProductSKU</code>\n            </p>\n
        </li>\n        <li>\n        <p>\n            <code>Status</code>\n        </p>\n        </li>\n
<li>\n        <p>\n            <code>Fingerprint</code>\n        </p>\n        </li>\n        <li>\n
<p>\n            <code>IssuerName</code>\n        </p>\n        </li>\n        <li>\n        <p>\n
<code>Beneficiary</code>\n        </p>\n        </li>\n        </ul>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
        "type": "structure",
```

```
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#GrantedLicenseList",
          "traits": {
            "smithy.api#documentation": "<p>Received license details.</p>"
          }
        },

        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListResourceInventory": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FailedDependencyException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#InventoryFilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n          <li>\n            <p>\n              <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
          </li>\n          <li>\n            <p>\n              <code>application_name</code> - The name of the
application. \n          Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
   </li>\n          <li>\n            <p>\n              <code>license_included</code> - The type of license included. \n
       Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n          Possible values
are <code>sql-server-enterprise</code> | \n          <code>sql-server-standard</code> | \n          <code>sql-
server-web</code> | \n          <code>windows-server-datacenter</code>.</p>\n          </li>\n          <li>\n
   <p>\n              <code>platform</code> - The platform of the resource.\n          Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n          </li>\n          <li>\n            <p>\n
        <code>resource_id</code> - The ID of the resource. \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n          </li>\n          <li>\n            <p>\n
   <code>tag:<key></code> - The key/value combination of a tag assigned \n          to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n          <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n          </li>\n        </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

```
"com.amazonaws.licensemanager#ListResourceInventoryResponse": {
    "type": "structure",
    "members": {
        "ResourceInventoryList": {
            "target": "com.amazonaws.licensemanager#ResourceInventoryList",
            "traits": {
                "smithy.api#documentation": "<p>Information
about the resources.</p>"
            }
        },
        "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#ListTagsForResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
```

```
                }
          },
      "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
              "ResourceArn": {
                  "target": "com.amazonaws.licensemanager#String",
                  "traits": {
                      "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

      "smithy.api#required": {}
                  }
              }
          },
          "traits": {
              "smithy.api#input": {}
          }
      },
      "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
              "Tags": {
                  "target": "com.amazonaws.licensemanager#TagList",
                  "traits": {
                      "smithy.api#documentation": "<p>Information about the tags.</p>"
                  }
              }
          },
          "traits": {
              "smithy.api#output": {}
          }
      },
      "com.amazonaws.licensemanager#ListTokens": {
          "type": "operation",
          "input": {
              "target": "com.amazonaws.licensemanager#ListTokensRequest"
          },
          "output": {
              "target": "com.amazonaws.licensemanager#ListTokensResponse"
          },
          "errors":
[
              {
                  "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
                  "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists your tokens.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTokensRequest": {
      "type": "structure",
      "members": {
        "TokenIds": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {

            "smithy.api#documentation": "<p>Token IDs.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n          <li>\n            <p>\n              <code>LicenseArns</code>\n            </p>\n          </li>\n
</ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
```

```
          "smithy.api#input": { }
        }
      },
      "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
          "Tokens": {
            "target": "com.amazonaws.licensemanager#TokenList",
            "traits": {
              "smithy.api#documentation": "<p>Received token details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": { }
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n        consumption details by resource at a selected point in time. Use this action to audit the\n        current
license consumption for any license inventory and configuration.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanager#Filters",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n
<li>\n        <p>\n                <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n        <p>\n
    <code>resourceAccount</code> - The ID of the account that owns the resource. \n        Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        </ul>"
                }
            }
```

```
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {

"LicenseConfigurationUsageList": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#Location": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Long": {
      "type": "long"
    },
    "com.amazonaws.licensemanager#ManagedResourceSummary": {
      "type": "structure",
      "members": {
        "ResourceType":
{
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
          }
        },
        "AssociationCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
          }
        }
```

```
      },
      "traits": {
        "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
      }
    },
    "com.amazonaws.licensemanager#MaxSize100": {

      "type": "integer",
      "traits": {
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "com.amazonaws.licensemanager#MaxSize3StringList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 3
        }
      }
    },
    "com.amazonaws.licensemanager#Message": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Metadata": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The key name.</p>"
          }
        },

        "Value": {
          "target": "com.amazonaws.licensemanager#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>The value.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
      }
    },
    "com.amazonaws.licensemanager#MetadataList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Metadata"
      }
    },
    "com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#Options": {
      "type": "structure",
      "members": {
        "ActivationOverrideBehavior": {
          "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
          "traits": {
            "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n          Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n          <ul>\n          <li>\n              <p>As a license administrator, you can
optionally specify an\n              <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
  </li>\n          <li>\n            <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n              account in your
organization.</p>\n          </li>\n          <li>\n              <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n              <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n            are activating the grant.</p>\n          </li>\n          </ul>\n          <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n            <dd>\n              <p>Use this value to activate a grant
without replacing any member accounts\n              active grants for the same product.</p>\n          </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n            <dd>\n              <p>Use this value to activate a
```

grant and disable other active grants in any\n   member accounts
 for the same product. This action will also replace their\n   previously activated grants with this activated
grant.</p>\n   </dd>\n   </dl>"
                }
              }
            },
            "traits": {
                "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n   override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "com.amazonaws.licensemanager#OrganizationConfiguration": {
            "type": "structure",
            "members": {
                "EnableIntegration": {
                    "target": "com.amazonaws.licensemanager#Boolean",
                    "traits": {
                        "smithy.api#default": false,
                        "smithy.api#documentation":
 "<p>Enables Organizations integration.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
            }
        },
        "com.amazonaws.licensemanager#PrincipalArnList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#Arn"
            },
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 1
                }
            }
        },
        "com.amazonaws.licensemanager#ProductInformation": {
            "type": "structure",
            "members": {
                "ResourceType": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {

```
                    "smithy.api#documentation": "<p>Resource type. The
 possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductInformationFilterList": {
                "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
                "traits": {
                    "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n        <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n        <p>The following filters and are supported when
the resource type \n        is <code>SSM_MANAGED</code>:</p>\n
     <ul>\n         <li>\n             <p>\n                 <code>Application Name</code> - The name of the
application.\n           Logical operator is <code>EQUALS</code>.</p>\n         </li>\n         <li>\n
<p>\n                 <code>Application Publisher</code> - The publisher of the application.\n           Logical
operator is <code>EQUALS</code>.</p>\n         </li>\n         <li>\n             <p>\n
<code>Application Version</code> - The version of the application.\n           Logical operator is
<code>EQUALS</code>.</p>\n         </li>\n         <li>\n             <p>\n                 <code>Platform
Name</code> - The name of the platform.\n           Logical operator is <code>EQUALS</code>.</p>\n
</li>\n         <li>\n             <p>\n                 <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
     </li>\n         <li>\n             <p>\n                 <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>.  The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n         </p>\n         </li>\n         <li>\n             <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n           Logical operator is <code>NOT_EQUALS</code>.</p>\n         </li>\n
<li>\n             <p>\n                 <code>License Included</code> - The type of license included.\n           Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n           Possible values are:
 <code>sql-server-enterprise</code> | \n             <code>sql-server-standard</code> | \n             <code>sql-server-
web</code> | \n             <code>windows-server-datacenter</code>.</p>\n         </li>\n         </ul>\n
<p>The following filters and logical operators are supported when the resource type\n         is
<code>RDS</code>:</p>\n         <ul>\n         <li>\n             <p>\n                 <code>Engine Edition</code> -
The edition of the database engine.\n           Logical operator is <code>EQUALS</code>.\n           Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n         </li>\n         <li>\n             <p>\n                 <code>License Pack</code> - The license
pack.\n           Logical operator is <code>EQUALS</code>.\n           Possible values are: <code>data
guard</code> | \n             <code>diagnostic pack sqlt</code> | \n
     <code>tuning pack sqlt</code> | \n             <code>ols</code> | \n             <code>olap</code>.</p>\n
 </li>\n         </ul>",
                    "smithy.api#required": {}
                }
            }
        },
```

```
      "traits": {
        "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilter": {
      "type": "structure",
      "members": {
        "ProductInformationFilterName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Filter name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductInformationFilterValue": {
          "target": "com.amazonaws.licensemanager#StringList",

        "traits": {
            "smithy.api#documentation": "<p>Filter value.</p>"
          }
        },
        "ProductInformationFilterComparator": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Logical operator.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes product information filters.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformationFilter"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformation"
      }
    },
    "com.amazonaws.licensemanager#ProvisionalConfiguration": {
      "type": "structure",
      "members": {
```

```
    "MaxTimeToLiveInMinutes": {
      "target": "com.amazonaws.licensemanager#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
}
},
"com.amazonaws.licensemanager#RateLimitExceededException": {
  "type": "structure",
  "members": {
    "Message": {
      "target": "com.amazonaws.licensemanager#Message"
    }
  },
  "traits": {
    "aws.protocols#awsQueryError": {
      "code": "RateLimitExceeded",
      "httpResponseCode": 429
    },
    "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 429
  }
},
"com.amazonaws.licensemanager#ReceivedMetadata": {
  "type": "structure",
  "members": {
    "ReceivedStatus": {
      "target": "com.amazonaws.licensemanager#ReceivedStatus",
      "traits": {
        "smithy.api#documentation": "<p>Received status.</p>"
      }
    },
    "ReceivedStatusReason": {
      "target": "com.amazonaws.licensemanager#StatusReasonMessage",
      "traits": {
        "smithy.api#documentation": "<p>Received
status reason.</p>"
      }
    },
    "AllowedOperations": {
      "target": "com.amazonaws.licensemanager#AllowedOperationList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Allowed operations.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
        }
      },
      "com.amazonaws.licensemanager#ReceivedStatus": {
        "type": "enum",
        "members": {
          "PENDING_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_WORKFLOW"
            }
          },
          "PENDING_ACCEPT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue":
"PENDING_ACCEPT"
            }
          },
          "REJECTED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "REJECTED"
            }
          },
          "ACTIVE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ACTIVE"
            }
          },
          "FAILED_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED_WORKFLOW"
            }
          },
          "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DELETED"
            }
```

```
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
           "traits": {
             "smithy.api#enumValue": "DISABLED"
           }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
             "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    }
  },
  "com.amazonaws.licensemanager#RedirectException": {
    "type": "structure",
    "members": {
      "Location": {
        "target": "com.amazonaws.licensemanager#Location",
        "traits": {
           "smithy.api#httpHeader": "Location"
        }
      },
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
       "smithy.api#error": "client",
      "smithy.api#httpError": 308
    }
  },
  "com.amazonaws.licensemanager#RejectGrant": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#RejectGrantRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#RejectGrantResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
              },
              {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
              },
              {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              },
              {
                "target": "com.amazonaws.licensemanager#ValidationException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
            }
          },
          "com.amazonaws.licensemanager#RejectGrantRequest": {
            "type": "structure",
            "members": {
              "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                  "smithy.api#required": {}
                }
              }
            },
            "traits": {
              "smithy.api#input": {}

          }
          },
          "com.amazonaws.licensemanager#RejectGrantResponse": {
            "type": "structure",
            "members": {
              "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                  "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
              },
              "Status": {
```

```
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#RenewType":
{
        "type": "enum",
        "members": {
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "WEEKLY": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Weekly"
            }
          },
          "MONTHLY": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Monthly"
            }
          }
        }
    },
    "com.amazonaws.licensemanager#ReportContext": {
      "type": "structure",
      "members": {
        "licenseConfigurationArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
```

```
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportFrequency": {
        "type": "structure",
        "members": {
            "value": {
                "target": "com.amazonaws.licensemanager#Integer",
                "traits": {
                    "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n        The only supported value is <code>1</code>.</p>"
                }
            },
            "period": {
                "target": "com.amazonaws.licensemanager#ReportFrequencyType",

                "traits": {
                    "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportFrequencyType": {
        "type": "enum",
        "members": {
            "DAY": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "DAY"
                }
            },
            "WEEK": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "WEEK"
                }
            },
```

```
        "MONTH": {
          "target": "smithy.api#Unit",
          "traits":
{
            "smithy.api#enumValue": "MONTH"
          }
        }
      }
    }
  },
  "com.amazonaws.licensemanager#ReportGenerator": {
    "type": "structure",
    "members": {
      "ReportGeneratorName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Name of the report generator.</p>"
        }
      },
      "ReportType": {
        "target": "com.amazonaws.licensemanager#ReportTypeList",
        "traits": {
          "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
        }
      },
      "ReportContext": {
        "target": "com.amazonaws.licensemanager#ReportContext",
        "traits": {
          "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
        }
      },
      "ReportFrequency": {
        "target": "com.amazonaws.licensemanager#ReportFrequency",
        "traits": {
          "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
        }
      },
      "LicenseManagerReportGeneratorArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
        }
      },
      "LastRunStatus": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
        }
```

            },
            "LastRunFailureReason":

 {

                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
                }
            },
            "LastReportGenerationTime": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
                }
            },
            "ReportCreatorAccount": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report generator.</p>"
                }
            },
            "Description": {
                "target": "com.amazonaws.licensemanager#String",

        "traits": {
                    "smithy.api#documentation": "<p>Description of the report generator.</p>"
                }
            },
            "S3Location": {
                "target": "com.amazonaws.licensemanager#S3Location",
                "traits": {
                    "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published to.</p>"
                }
            },
            "CreateTime": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Time the report was created.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                    "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
                }

```
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
    }
  },
  "com.amazonaws.licensemanager#ReportGeneratorList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ReportGenerator"
    }
  },
  "com.amazonaws.licensemanager#ReportGeneratorName": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "com.amazonaws.licensemanager#ReportType": {
    "type": "enum",
    "members": {
      "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
        }

      },
      "LICENSE_CONFIGURATION_USAGE_REPORT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "LicenseConfigurationUsageReport"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#ReportTypeList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ReportType"
    }
  },
  "com.amazonaws.licensemanager#ResourceInventory": {
    "type": "structure",
    "members": {
```

```
      "ResourceId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits":
{
          "smithy.api#documentation": "<p>Type of resource.</p>"
        }
      },
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "Platform": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Platform of the resource.</p>"
        }
      },
      "PlatformVersion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
        }
      },
      "ResourceOwningAccountId":
{
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about a resource.</p>"
    }
  },
  "com.amazonaws.licensemanager#ResourceInventoryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#ResourceInventory"
    }
```

```
    },
    "com.amazonaws.licensemanager#ResourceLimitExceededException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "ResourceLimitExceeded",

                "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "InvalidResource.NotFound",
                "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>The resource cannot be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#ResourceType": {
        "type": "enum",
        "members": {
            "EC2_INSTANCE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "EC2_INSTANCE"
                }
            },
            "EC2_HOST": {
                "target": "smithy.api#Unit",
                "traits": {
```

```
              "smithy.api#enumValue": "EC2_HOST"
            }
          },
          "EC2_AMI": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "EC2_AMI"
            }
          },
          "RDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "RDS"
            }
          },
          "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#S3Location": {
        "type": "structure",
        "members": {
          "bucket": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
            }
          },
          "keyPrefix": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
        }
      },
      "com.amazonaws.licensemanager#ServerInternalException": {
        "type": "structure",
        "members": {
          "Message": {
```

```json
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InternalError",
            "httpResponseCode": 500
          },
          "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanager#SignedToken": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 4096
          }
        }
      },
      "com.amazonaws.licensemanager#StatusReasonMessage": {
        "type": "string",
        "traits":
{
          "smithy.api#length": {
            "min": 0,
            "max": 400
          },
          "smithy.api#pattern": "^[\\s\\S]+$"
        }
      },
      "com.amazonaws.licensemanager#String": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#StringList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "com.amazonaws.licensemanager#Tag": {
        "type": "structure",
        "members": {
          "Key": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Tag key.</p>"
```

```
              }
            },
            "Value": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation":
"<p>Tag value.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#TagKeyList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#String"
          }
        },
        "com.amazonaws.licensemanager#TagList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Tag"
          }
        },
        "com.amazonaws.licensemanager#TagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#TagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#TagResourceResponse"
          },
          "errors": [
            {
              "target":
"com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
```

```
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits":
{
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>One or more tags.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#TokenData": {
        "type": "structure",
        "members": {
          "TokenId": {
            "target": "com.amazonaws.licensemanager#String",
             "traits": {
              "smithy.api#documentation": "<p>Token ID.</p>"
            }
          },
```

```
        "TokenType": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "ExpirationTime": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
 format.</p>"
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
            "smithy.api#documentation": "<p>Data specified by the caller.</p>"
          }
        },
        "RoleArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
          }
        }
      },
      "traits": {

        "smithy.api#documentation": "<p>Describes a token.</p>"
      }
    },
    "com.amazonaws.licensemanager#TokenList": {
      "type": "list",
```

```
      "member": {
        "target": "com.amazonaws.licensemanager#TokenData"
      }
    },
    "com.amazonaws.licensemanager#TokenString": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 4096
        },
        "smithy.api#pattern": "^\\S+$"
      }
    },
    "com.amazonaws.licensemanager#TokenType": {
      "type": "enum",
      "members": {
        "REFRESH_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REFRESH_TOKEN"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
{
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UntagResourceResponse"
      },
```

```
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
          "smithy.api#required": {}
        }
      },
      "TagKeys": {
        "target": "com.amazonaws.licensemanager#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#UntagResourceResponse": {
    "type": "structure",
    "members": {},
```

```
        "traits": {
          "smithy.api#output": {}
        }
      },
    "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
      },
      "output": {

"target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
```

```
"LicenseConfigurationStatus": {
    "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
    "traits": {
        "smithy.api#documentation": "<p>New status of the license configuration.</p>"
    }
},
"LicenseRules": {
    "target": "com.amazonaws.licensemanager#StringList",
    "traits": {

    "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
    configuration is licenseAffinityToHost.</p>"
    }
},
"LicenseCount": {
    "target": "com.amazonaws.licensemanager#BoxLong",
    "traits": {
        "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
    }
},
"LicenseCountHardLimit": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
    "traits": {
        "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
    }
},
"Name": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
    }
},
"Description": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>New description of the license configuration.</p>"
    }
},
"ProductInformationList": {
    "target": "com.amazonaws.licensemanager#ProductInformationList",
    "traits": {
        "smithy.api#documentation": "<p>New product information.</p>"
    }
},
"DisassociateWhenNotFound": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            }
        },
        "traits":
{
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
```

```
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
            "smithy.api#required": {}
          }
        },
        "ReportGeneratorName": {
          "target": "com.amazonaws.licensemanager#ReportGeneratorName",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>",
            "smithy.api#required": {}
          }
        },
        "Type": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n        <ul>\n        <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n        </li>\n        <li>\n            <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n        </li>\n        </ul>",
            "smithy.api#required": {}
          }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>The report context.</p>",
            "smithy.api#required": {}
          }
        },
        "ReportFrequency": {
          "target": "com.amazonaws.licensemanager#ReportFrequency",
```

```
                    "traits":
{
                        "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ClientToken": {
                    "target": "com.amazonaws.licensemanager#ClientRequestToken",
                    "traits": {
                        "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Description": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Description of the report generator.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
```

```
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },

        {
          "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
        },
        {
          "target": "com.amazonaws.licensemanager#LicenseUsageException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
      }
    },

    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
            "smithy.api#required": {}
          }
        },
        "AddLicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
          }
        },
        "RemoveLicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation":
"<p>ARNs of the license configurations to remove.</p>"
          }
```

```
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
            }
          },
          "SnsTopicArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
            }
          },
          "OrganizationConfiguration": {
            "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
            }
          },
          "EnableCrossAccountsDiscovery": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

        }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#UsageOperation": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          }
        }
      },
      "com.amazonaws.licensemanager#ValidationException": {
```

```
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

    "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {
```

```go
	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// Current version
	of the license.
	//
	// This member is required.
	SourceVersion *string

	noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

	// Date when the license is deleted.
	DeletionDate *string

	// License status.
	Status types.LicenseDeletionStatus

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack,
	options); err != nil {
		return err
```

```
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
     return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
     return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
     return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
     return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
     return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
     return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
     return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
     return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
     return err
    }
    if
    err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
     return err
    }
    if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
     return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
    middleware.Before); err != nil {
     return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
     return err
    }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:        region,
 ServiceID:     ServiceID,
 OperationName: "DeleteLicense",
 }
}
```

1.35 aws-aws-sdk-go-v2 1.25.10

1.35.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
```

```go
	if params == nil {
		params = &AssociateLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
	if err != nil {
		return nil,
	err
	}

	out := result.(*AssociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type AssociateLicenseInput struct {

	// The type of license to associate with the workspace.
	//
	// This member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to associate the license with.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

	// A structure containing data about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
```

```go
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err !=
nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack);
err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack,
 options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

```
name: License Scan

on: [pull_request]

jobs:
 licensescan:
```

```
name: License Scan
runs-on: ubuntu-latest
strategy:
  matrix:
    python-version: [3.9]


steps:
  - name: Checkout target
    uses: actions/checkout@v2
    with:
      path: sdkbase
      ref: ${{ github.base_ref }}
  - name: Checkout this ref
    uses: actions/checkout@v2
    with:
      path: new-ref
      fetch-depth: 0
  - name: Get Diff
    run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }} ${{ github.sha }} > refDiffFiles.txt
  - name: Get Target Files
    run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
  - name: Checkout scancode
    uses: actions/checkout@v2
    with:
      repository: nexB/scancode-toolkit
      path: scancode-toolkit
      fetch-depth: 1
  - name: Set up Python ${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```go
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary
 *string

 noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
```

```go
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
```

```go
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
		return
err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckInLicense",
	}
}
```

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
```

```json
        "type": "integer"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n     Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {

          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
```

```
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
        "type": "structure",
        "members": {}
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n      Organizations integration has been enabled, and
 which Regions data will be aggregated from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
```

```
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
          "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
          }
        },
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
          }
        },
        "InstanceID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
          }
        },
        "InstanceType": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance type of the resource.</p>"
          }

      },
        "AccountID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
```

```
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
            }
          },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
              "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
          },
          "LastUpdatedTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
```

```
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
          }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
              "message": {
                  "target": "smithy.api#String"
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
              "smithy.api#error": "server"
          }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
          "type": "service",
          "version": "2018-05-10",
          "operations": [
              {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
              },
              {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
              },
              {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
              },
              {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
              }
          ],

          "traits": {
              "aws.api#service": {
                  "sdkId": "License Manager Linux Subscriptions",
                  "arnNamespace": "license-manager-linux-subscriptions",
                  "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
              },
              "aws.auth#sigv4": {
                  "name": "license-manager-linux-subscriptions"
              },
              "aws.protocols#restJson1": {},
              "smithy.api#cors": {
                  "additionalAllowedHeaders": [
                      "*",
                      "content-type",
                      "x-amz-content-sha256",
```

```
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                "x-api-key",
                "authorization",
                "x-amz-date",

    "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
            ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
                "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
                    "documentation": "The
 AWS region used to dispatch the request.",
                    "type": "String"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
```

```
          "type": "Boolean"
        },
      "Endpoint": {
          "builtIn": "SDK::Endpoint",
          "required": false,
          "documentation": "Override the endpoint used to send this request",
          "type": "String"
        }
      },
    "rules": [
        {
          "conditions": [
              {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                  ]
              }
          ],
          "type": "tree",
          "rules": [
              {

            "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                      ]
                  }
                ],
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
              "type": "error"
            },
            {
              "conditions": [],
              "type": "tree",
              "rules": [
                  {

                "conditions": [
                        {
                            "fn": "booleanEquals",
```

```
                    "argv": [
                       {
                          "ref": "UseDualStack"
                       },
                       true
                    ]
                 }
              ],
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
              "type": "error"
           },
           {
              "conditions": [],

      "endpoint": {
                    "url": {
                       "ref": "Endpoint"
                    },
                    "properties": {},
                    "headers": {}
                 },
                 "type": "endpoint"
              }
           ]
        }
     ]
  },
  {
     "conditions": [],
     "type": "tree",
     "rules": [
        {
           "conditions": [
              {
                 "fn": "isSet",

"argv": [
                    {
                       "ref": "Region"
                    }
                 ]
              }
           ],
           "type": "tree",
           "rules": [
              {
                 "conditions": [
                    {
```

```
                    "fn": "aws.partition",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ],

    "assign": "PartitionResult"
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      },

        {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [
                          {

                      "fn": "booleanEquals",
                            "argv": [
                              true,
                              {
                                "fn": "getAttr",
                                "argv": [
                                  {
                                      "ref": "PartitionResult"
```

```
                },
                "supportsFIPS"
              ]

  }
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },

                    "supportsDualStack"
                        ]
                      }
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [],
                    "type": "tree",
                    "rules": [
                      {

                  "conditions": [],
                            "endpoint": {
                              "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                              "properties": {},
                              "headers": {}
                            },
                            "type": "endpoint"
                          }
                    ]
                  }
                ]

              },
                {
                  "conditions": [],
```

                                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

                                "type": "error"
                              }
                          ]
                    },
                    {
                        "conditions": [
                          {
                              "fn": "booleanEquals",
                              "argv": [
                                  {

                "ref": "UseFIPS"
                                  },
                                  true
                              ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
                          {
                              "conditions": [
                                  {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,

            {

                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsFIPS"
                                            ]
                                        }
                                    ]
                                  }
                              ],
                              "type": "tree",

                "rules": [
                                  {
                                    "conditions": [],
                                    "type": "tree",
                                    "rules": [
                                        {

```
                                    "conditions": [],
                                    "endpoint": {
                                      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                        "properties": {},
                                        "headers": {}


                                    },
                                     "type": "endpoint"
                                  }
                               ]
                             }
                          ]
                       },
                       {
                          "conditions": [],
                          "error": "FIPS is enabled but this partition does not support FIPS",
                          "type": "error"
                       }
                     ]

                },
                                {
                                   "conditions": [
                                     {
                                        "fn": "booleanEquals",
                                        "argv": [
                                          {
                                             "ref": "UseDualStack"
                                          },
                                           true
                                        ]
                                     }
                                   ],
                                   "type": "tree",
                                   "rules": [
                                     {

                                   "conditions": [
                                        {
                                           "fn": "booleanEquals",
                                           "argv": [
                                             true,
                                             {
                                                "fn": "getAttr",
                                                "argv": [
                                                  {
                                                     "ref": "PartitionResult"
```

```
                },
                  "supportsDualStack"
                ]
              }
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [],
            "type": "tree",
            "rules": [
              {
                "conditions": [],

                "endpoint": {
                  "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ]
          }
        ]
      },

  {
                "conditions": [],
                "error": "DualStack is enabled but this partition does not support DualStack",
                "type": "error"
              }
            ]
          },
          {
            "conditions": [],
            "type": "tree",
            "rules": [
              {
                "conditions": [],
                "endpoint": {
                  "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
```

```
                                "type": "endpoint"
                              }
                            ]
                          }
                        ]
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "error": "Invalid Configuration: Missing Region",
                    "type": "error"
                  }
                ]
              }
            ]
          },
          "smithy.rules#endpointTests": {
            "testCases": [
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {

"endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
         "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {

    "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
```

```
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint":
```
{
```
                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-gov-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
```

```
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint":

{

                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
             "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
      },
      "params": {

      "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
```

```
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled but this partition does not support DualStack"
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",

    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://example.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
```

```
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                  }
              },
              {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
                    "endpoint": {
                      "url": "https://example.com"
                    }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
      "Endpoint": "https://example.com"
                }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "Missing region",
              "expect": {
```

```
                        "error": "Invalid Configuration: Missing Region"
                    }

                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type":
"operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with
commercial Linux\n      subscriptions.</p>",
          "smithy.api#http": {

      "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by
```

the name of\n   <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n   <p>The valid names for this filter are:</p>\n   \n   \n   <p>\n   <code>AmiID</code>\n   </p>\n   \n   \n   <p>\n   <code>InstanceID</code>\n   </p>\n   \n \n   <p>\n   <code>AccountID</code>\n   </p>\n   \n   \n <p>\n   <code>Status</code>\n   </p>\n   \n   \n   <p>\n <code>Region</code>\n   </p>\n   \n   \n   <p>\n <code>UsageOperation</code>\n   </p>\n   \n   \n   <p>\n   <code>ProductCode</code>\n   </p>\n   \n   \n   <p>\n   <code>InstanceType</code>\n   </p>\n   \n   \n   <p>The valid Operators for this filter are:</p>\n   \n   \n   <p>\n   <code>contains</code>\n   </p>\n \n   \n   <p>\n   <code>equals</code>\n   </p>\n   \n   \n   <p>\n   <code>Notequal</code>\n   </p>\n   \n   "
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
            "Instances": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                "traits": {
                    "smithy.api#documentation":
"<p>An array that contains instance objects.</p>"
                }
            },

```
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n     organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }

  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
```

```
"members": {
  "Filters": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n      <ul>\n
<li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n
<p>The valid Operators for this filter are:</p>\n
    <ul>\n      <li>\n        <p>\n            <code>contains</code>\n          </p>\n        </li>\n
<li>\n        <p>\n            <code>equals</code>\n          </p>\n        </li>\n        <li>\n          <p>\n
    <code>Notequal</code>\n          </p>\n        </li>\n      </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Token for the next set of results.</p>",
      "smithy.api#length": {

        "min": 1,
          "max": 16384
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
```

```
                            "target": "smithy.api#String",
                            "traits": {
                                "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                            }
                        }
                    }
                }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
                "type": "string",
                "traits": {
                    "smithy.api#enum": [
                        {
                            "name": "EQUAL",
                            "value": "Equal",
                            "documentation": "Equal operator"
                        },
                        {
                            "name": "NOT_EQUAL",
                            "value": "NotEqual",
                            "documentation": "Not equal operator"
                        },
                        {
                            "name": "CONTAINS",
                            "value": "Contains",
                            "documentation": "Contains operator"
                        }
                    ],
                    "smithy.api#length": {
                        "min": 1,
                        "max": 20
                    }
                }
            },

          "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
                "type": "string",
                "traits": {
                    "smithy.api#enum": [
                        {
                            "name": "Enabled",
                            "value": "Enabled",
                            "documentation": "Enabled OrganizationIntegration"
                        },
                        {
                            "name": "Disabled",
                            "value": "Disabled",
                            "documentation": "Disabled OrganizationIntegration"
```

```
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [

                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
                    "value": "Successful",
                    "documentation": "Successful status"
                },
                {
                    "name": "Failed",
                    "value": "Failed",
                    "documentation": "Failed status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits":
{
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
```

```
            }
          },
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 100
            },
            "smithy.api#uniqueItems": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
          "type": "map",
          "key": {
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"

              }
            },
            "Type": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
              }
            },
            "InstanceCount": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux
```

subscription.</p>"
              }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
              "type": "list",
              "member": {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
              }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
              "type": "structure",
              "members": {
                  "message": {
                      "target": "smithy.api#String"
                  }
              },
              "traits": {
                  "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
                  "smithy.api#error": "client"
              }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
              "type": "operation",
              "input": {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
              },

              "output": {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
              },
              "errors": [
                  {
                      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
                  },
                  {
                      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                  },
                  {
                      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                  }
              ],
              "traits": {
                  "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
                  "smithy.api#http": {
                      "code": 200,
                      "method": "POST",
                      "uri": "/subscription/UpdateServiceSettings"
                  },

```
                    "smithy.api#idempotent": {}
                }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
                "type": "structure",
                "members": {
                    "LinuxSubscriptionsDiscovery": {
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                        "traits": {
                            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                            "smithy.api#required": {}
                        }
                    },
                    "LinuxSubscriptionsDiscoverySettings": {
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                        "traits": {
                            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",

    "smithy.api#required": {}
                        }
                    },
                    "AllowUpdate": {
                        "target": "smithy.api#Boolean",
                        "traits": {
                            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                        }
                    }
                }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
                "type": "structure",
                "members": {
                    "LinuxSubscriptionsDiscovery": {
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                        "traits": {
                            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                        }
                    },

            "LinuxSubscriptionsDiscoverySettings": {
                        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                        "traits": {
                            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
```

settings include if Organizations\n   integration has been enabled, and which Regions data will be aggregated from.</p>"
```
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation":
 "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```go
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
	if params == nil {
		params = &ListLicensesInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*ListLicensesOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type ListLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//   - Beneficiary
	//   - ProductSKU
	//   - Fingerprint
	//
	//   - Status
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListLicensesOutput struct {
```

```go
	// License details.
	Licenses []types.License

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	if
	err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
```

```go
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
	err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:      region,
		ServiceID:
```

```go
      ServiceID,
   OperationName: "ListLicenses",
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 //
 This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string
```

```go
  noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "DisassociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```go
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
	middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
```

```go
  }
  return nil
 }

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
 {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type
```

```
CheckoutLicenseInput struct {

    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string
```

```go
    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource
    Name (ARN) of the checkout license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil
    {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
```

```go
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack);
err != nil {
return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
```

```go
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```go
type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier
    that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see RFC 7518 Digital Signature with
    // RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string
```

```go
    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }
```

```go
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
if err = addClientUserAgent(stack, options); err != nil {
 return
err
 }
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
if err = addRequestResponseLogging(stack, options); err != nil {
```

```go
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
 params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```go
type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide
    to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

    // Product name.
    //
    // This member is required.
    ProductName *string

    // Product SKU.
    //
```

```go
	// This member is required.
	ProductSKU *string

	// Date and time range during which the license is valid, in ISO8601-UTC format.
	//
	// This member is required.
	Validity
	*types.DatetimeRange

	// Information about the license.
	LicenseMetadata []types.Metadata

	noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
		return fmt.Errorf("add
	protocol finalizers: %v", err)
	}
```

```go
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
return err
}
```

```go
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:       region,
		ServiceID:     ServiceID,
		OperationName:
		"CreateLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
	if params == nil {
		params = &GetLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
```

```go
c.addOperationGetLicenseMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*GetLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type GetLicenseInput struct {

	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// License version.
	Version
	*string

	noSmithyDocumentSerde
}

type GetLicenseOutput struct {

	// License details.
	License *types.License

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
```

```go
	return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return
err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpGetLicenseValidationMiddleware(stack);
err != nil {
		return err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
```

```go
err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "GetLicense",
 }
}
```

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
 params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //  - ProductSKU
 //  - Status
 //  - Fingerprint
 //  - IssuerName
```

```go
    //   - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
    return err
   }
   if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
   }
   if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
    return err
   }
   if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
    return err
   }
   if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
   }
   if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
   }
   if err = addRetryMiddlewares(stack, options); err != nil {
    return err
   }
   if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
   }
   if err = awsmiddleware.AddRecordResponseTiming(stack);
   err != nil {
    return err
   }
   if err = addClientUserAgent(stack, options); err != nil {
    return err
   }
   if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
   }
   if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
   }
   if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
   }
   if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
   middleware.Before); err != nil {
    return err
   }
   if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
   }
   if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
```

```
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
     return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
     return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
     return err
    }
    return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:      region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
```

                        Apache License
                     Version 2.0, January 2004
                   http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership
   of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this

License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

 4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding
those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
```

        },
        "traits": {
          "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
        "type": "structure",
        "members": {
          "DirectoryId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
"<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n        <note>\n          <p>Your estimated bill for charges on the number of users and related
costs will take 48\n       hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
        monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The identity provider of the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the user.</p>"
```

```
                }
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
            "type": "structure",
            "members": {
              "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                  "smithy.api#required": {}
                }
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
            "type": "integer"
          },
          "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
            "type": "structure",
            "members": {
              "message": {
                "target": "smithy.api#String"
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n    resource.</p>",
              "smithy.api#error": "server"
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
              },
```

```
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
      "type": "structure",
```

```
          "members": {
            "IdentityProviderSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
```

```
                    "uri": "/user/DisassociateUser"
                },
                "smithy.api#idempotent": {}
            }

    },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
            "type": "structure",
            "members": {
                "Username": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                        "smithy.api#required": {}
                    }
                },
                "InstanceId": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                        "smithy.api#required": {}
                    }
                },
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#documentation":
"<p>An object that specifies details for the identity provider.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Domain": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The domain name of the user.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
            "type": "structure",
            "members": {
                "InstanceUserSummary": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                    "traits": {
                        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                        "smithy.api#required": {}
```

```
            }
          }
        }
      },

  "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used
 to return more specific results from a describe\n     operation. Filters can be used to match a set of resources by
 specific criteria, such as tags,\n     attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
          }
        }
```

```
        },
        "traits": {
          "smithy.api#documentation":
"<p>Details about an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>An object that details the registered identity providers product
related configuration\n      settings such as the subnets to provision VPC endpoints.</p>",
            "smithy.api#required":
{}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
```

```
"<p>Describes an identity provider.</p>"
          }
        },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status
of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
```

```
          }
        },
      "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        },
        "AssociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
          }
        },
        "DisassociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
          }
        }

    },
        "traits": {
          "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
```

```
        "type": "service",

    "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },

        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
```

```
          },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
          "x-amzn-trace-id",
          "content-length",
          "x-api-key",
          "authorization",
          "x-amz-date",
          "x-amz-security-token",
          "Access-Control-Allow-Headers",
          "Access-Control-Allow-Methods",
          "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
          "x-amzn-errortype",
          "x-amzn-requestid",
          "x-amzn-trace-id"

]
      },
      "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
      "smithy.api#title": "AWS License Manager User Subscriptions",
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
          "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
          "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack
endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
```

```
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                    "type": "Boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "String"
                }

        },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },

                                        true
                                    ]
                                }
                            ],
                            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                            "type": "error"
                        },
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
```

```
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
              "type": "error"
          },
          {
            "conditions": [],
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },
              "properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ]
      },
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [

{
                "ref": "Region"
              }
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ],
```

```json
                    "assign": "PartitionResult"
                  }
                ],
                "type": "tree",

        "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"

                      },
                      true
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",

"argv": [

                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      },
```

```
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  true,
                                  {

        "fn": "getAttr",
                                    "argv": [
                                      {
                                        "ref": "PartitionResult"
                                      },
                                      "supportsDualStack"
                                    ]
                                  }
                                ]
                              }
                            ],
                            "type": "tree",
                            "rules": [
                              {

        "conditions": [],
                                "endpoint": {
                                  "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
                                "type": "endpoint"
                              }
                            ]
                          },
                          {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                            "type": "error"
                          }
                        ]
                      },
                      {
                        "conditions": [
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              {
                                "ref": "UseFIPS"
                              },
```

```
                        true
                      ]
                  }
              ],
              "type": "tree",

    "rules": [
                {
                    "conditions": [
                      {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                  {
                                      "ref": "PartitionResult"
                                  },
                                  "supportsFIPS"

                            ]
                          }
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                  {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}

        },
                        "type": "endpoint"
                    }
                  ]
              },
              {
                  "conditions": [],
                  "error": "FIPS is enabled but this partition does not support FIPS",
                  "type": "error"
              }
            ]
        },
```

```
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "type": "tree",
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {

              "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsDualStack"
                ]
            }
          ]
        }
      ],
      "type": "tree",
      "rules": [
        {

          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ]
    },
```

```
                                {
                                    "conditions": [],
                                    "error": "DualStack is enabled but this partition does
not support DualStack",
                                    "type": "error"
                                }
                            ]
                        },
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ]
                }
            ]
        },
        {
            "conditions":
[],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation":
```

"For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                    }
                },

            "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                    }
                },

```
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
        },

    "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
```

```json
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },

      "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
```

"params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                "expect":

{

                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                    }
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "error": "DualStack is enabled but this partition does not support DualStack"
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack":

true

                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"

```
                              }
                          },
                          "params": {
                              "Region": "us-iso-east-1",
                              "UseFIPS": false,
                              "UseDualStack": false
                          }
                      },
                      {
                          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
                          "expect": {
                              "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
                          },
                          "params": {
                              "Region": "us-isob-east-1",
                              "UseFIPS": true,
                              "UseDualStack": true
                          }
                      },
                      {
                          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                          "expect": {
                              "endpoint": {
                                  "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                              }
                          },
                          "params": {
                              "Region": "us-isob-east-1",
                              "UseFIPS": true,
                              "UseDualStack": false
                          }

                      },
                      {
                          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
                          "expect": {
                              "error": "DualStack is enabled but this partition does not support DualStack"
                          },
                          "params": {
                              "Region": "us-isob-east-1",
                              "UseFIPS": false,
                              "UseDualStack": true
                          }
                      },
                      {
                          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                          "expect": {
```

```
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                }
            },
            "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {

            "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
```

```
                    }
                },
                {
                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated":
{
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
```

```
            "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
            "smithy.api#required": {}
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
```

```
              "uri": "/instance/ListInstances"
            },
          "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "InstanceSummaries"
            }
          }
        },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
             "traits": {
               "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
             }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
            }
          }
        }
      },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type":
"structure",
        "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>Token for the next set of results.</p>"
```

```
                }
              }
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/user/ListProductSubscriptions"
                },
                "smithy.api#paginated": {
                    "inputToken": "NextToken",
                    "outputToken": "NextToken",
                    "pageSize": "MaxResults",
                    "items": "ProductUserSummaries"
```

```
                }
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
              "type": "structure",
              "members": {
                "Product": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                  }
                },
                "IdentityProvider": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                  "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                  }
                },
                "MaxResults": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                  "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                  }
                },
                "Filters": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                  "traits": {

    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n     more sets of key-value pairs that you specify.</p>"
                  }
                },
                "NextToken": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                  }
                }
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
              "type": "structure",
              "members": {
                "ProductUserSummaries": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
```

```
              "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
              }
          },

      "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
          }
      }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
```

```
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceUserSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n     more sets of key-value pairs that you specify.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {

            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
```

```
    },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for a user.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
          }
        },
        "SubscriptionEndDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                    "smithy.api#required": {}
                }
```

```
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n      subnets to provision VPC endpoints.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 404
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
      "type": "string",
```

```
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
        "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                    "min": 1
                },
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n     your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n     subnets to provision VPC endpoints, and the security group ID that
is associated with the VPC\n     endpoints. The security group should permit inbound TCP port 1688
communication from resources\n     in the VPC.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related costs
will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
 For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n        monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StartProductSubscription"
          }
        }
      },
```

```
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the user.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
```

```json
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
```

```
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required":
{}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the user.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation":
"<p>Metadata that describes the start product subscription operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
        }
```

```
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {

            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity\n     provider.</p>",
          "smithy.api#http": {
            "code": 200,
```

```json
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n     update any combination of settings in a single operation such as the:</p>\n
<ul>\n        <li>\n            <p>Subnets which you want to add to provision VPC endpoints.</p>\n        </li>\n
    <li>\n            <p>Subnets which you want to remove the VPC endpoints from.</p>\n        </li>\n
<li>\n            <p>Security group ID which permits traffic to the VPC endpoints.</p>\n        </li>\n        </ul>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
```

```
        "members": {
            "AddSubnets": {

                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n     require connectivity to activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID
that allows inbound TCP port 1688 communication between resources in\n     your VPC and the VPC endpoints for
activation servers.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as\n     the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
}
}
{
  "smithy": "2.0",
  "metadata": {
```

```
    "suppressions": [
      {
        "id": "HttpMethodSemantics",
        "namespace": "*"
      },
      {
        "id": "HttpResponseCodeSemantics",
        "namespace": "*"
      },
      {
        "id": "PaginatedTrait",
        "namespace": "*"
      },
      {
        "id": "HttpHeaderTrait",
        "namespace": "*"
      },
      {
        "id": "HttpUriConflict",
        "namespace": "*"
      },
      {
        "id": "Service",
        "namespace": "*"
      }
    ]
  },
  "shapes": {
    "com.amazonaws.licensemanager#AWSLicenseManager": {
      "type": "service",
      "version": "2018-08-01",
      "operations": [
        {
          "target": "com.amazonaws.licensemanager#AcceptGrant"
        },

  {
          "target": "com.amazonaws.licensemanager#CheckInLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CheckoutLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateGrant"
        },
```

```
    {
      "target": "com.amazonaws.licensemanager#CreateGrantVersion"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
    },
    {
      "target": "com.amazonaws.licensemanager#CreateToken"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicense"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#DeleteToken"
    },
    {
      "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
    }
},
    {
      "target": "com.amazonaws.licensemanager#GetAccessToken"
    },
    {
      "target": "com.amazonaws.licensemanager#GetGrant"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicense"
    }
```

        },
        {
          "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
        },
        {
          "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
        },
        {
          "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
        },
        {
          "target": "com.amazonaws.licensemanager#GetLicenseUsage"
        },
        {
          "target": "com.amazonaws.licensemanager#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
        },
        {
          "target": "com.amazonaws.licensemanager#ListDistributedGrants"
        },
        {
          "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
        },
        {
          "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
        },
        {
          "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
        },
        {
          "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
        },
        {
          "target": "com.amazonaws.licensemanager#ListLicenses"
        },
        {
          "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
        },
        {

          "target": "com.amazonaws.licensemanager#ListLicenseVersions"
        },
        {
          "target": "com.amazonaws.licensemanager#ListReceivedGrants"
        },
        {

            "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
          },
          {
            "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
          },
          {
            "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
          },
          {
            "target": "com.amazonaws.licensemanager#ListResourceInventory"
          },
          {
            "target": "com.amazonaws.licensemanager#ListTagsForResource"
          },
          {
            "target": "com.amazonaws.licensemanager#ListTokens"
          },
          {
            "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
           },
          {
            "target": "com.amazonaws.licensemanager#RejectGrant"
          },
          {
            "target": "com.amazonaws.licensemanager#TagResource"
          },
          {
            "target": "com.amazonaws.licensemanager#UntagResource"
          },
          {
            "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
          },
          {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
          },
          {
            "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
          },
          {
            "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
             "sdkId": "License Manager",
            "arnNamespace": "license-manager",
            "cloudFormationName": "LicenseManager",
            "cloudTrailEventSource": "licensemanager.amazonaws.com",

              "endpointPrefix": "license-manager"
            },
            "aws.auth#sigv4": {
              "name": "license-manager"
            },
            "aws.protocols#awsJson1_1": {},
            "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n   Amazon Web Services accounts and on-premises servers.</p>",
            "smithy.api#title": "AWS License Manager",
            "smithy.api#xmlNamespace": {
              "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
            },
            "smithy.rules#endpointRuleSet": {
              "version": "1.0",
              "parameters": {
                "Region": {
                  "builtIn": "AWS::Region",
                  "required": false,
                  "documentation": "The AWS region used to dispatch the request.",
                  "type": "String"
                },
                "UseDualStack": {
                  "builtIn": "AWS::UseDualStack",
                  "required": true,
                  "default": false,
                  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                  "type": "Boolean"
                },
                "UseFIPS": {
                  "builtIn": "AWS::UseFIPS",
                  "required": true,
                  "default": false,
                  "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
                  "type": "Boolean"
                },
                "Endpoint": {
                  "builtIn": "SDK::Endpoint",
                  "required": false,
                  "documentation": "Override the endpoint used to send this request",
                  "type": "String"
                }
              },
              "rules": [
                {
                  "conditions": [

```json
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

              }
            ],
            "type": "tree",
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {

  "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                      "ref": "Endpoint"
```

```
                },
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ]
        },
        {
          "conditions": [
            {
              "fn": "isSet",
              "argv": [
                {
                  "ref": "Region"
                }
              ]
            }
          ],
          "type": "tree",
          "rules": [
            {
              "conditions": [
                {
                  "fn": "aws.partition",
                  "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref":
"UseFIPS"
                        },
                        true
                      ]
                    },
                    {
```

```
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                true,
                {
                  "fn": "getAttr",
                  "argv": [
                    {
                      "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                  ]
                }
              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                true,
                {
                  "fn": "getAttr",
                  "argv": [
                    {
                      "ref": "PartitionResult"
                    },

"supportsDualStack"

                  ]
                }
              ]
            }
          ],
          "type": "tree",
          "rules": [
```

```
                              {
                                 "conditions": [],
                                 "endpoint": {
                                    "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                       "properties": {},
                                       "headers": {}

                     },
                                    "type": "endpoint"
                                 }
                           ]
                        },
                        {
                           "conditions": [],
                           "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                           "type": "error"
                        }
                     ]
                  },
                  {
                     "conditions": [
                        {
                           "fn": "booleanEquals",

                   "argv": [
                              {
                                 "ref": "UseFIPS"
                              },
                              true
                           ]
                        }
                     ],
                     "type": "tree",
                     "rules": [
                        {
                           "conditions": [
                              {
                                 "fn": "booleanEquals",
                                 "argv": [
                                    true,

                 {

                                    "fn": "getAttr",
                                    "argv": [
                                       {
                                          "ref": "PartitionResult"
```

```
                },
                  "supportsFIPS"
              ]
          }
        ]
      }
    ],
    "type": "tree",
    "rules": [

{
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ]
    },
    {
      "conditions": [],
      "error": "FIPS is enabled but this
partition does not support FIPS",
      "type": "error"
    }
  ]
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "type": "tree",
   "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
```

```
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },

        "supportsDualStack"

                                        ]
                                    }
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}

                    },
                                "type": "endpoint"
                            }
                        ]
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ]
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ]
    }
]
```

```
          },
          {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
          }
        ]
      },
      "smithy.rules#endpointTests": {
        "testCases": [
          {
            "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint":
{

                  "url": "https://license-manager.af-south-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "af-south-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.ap-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "ap-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager.ap-northeast-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "ap-northeast-1",
              "UseFIPS": false,
              "UseDualStack": false
```

```
              }
            },
            {
              "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.ap-northeast-2.amazonaws.com"
                }
              },
              "params": {
                "Region": "ap-northeast-2",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.ap-northeast-3.amazonaws.com"
                }
              },
              "params": {
                "Region": "ap-northeast-3",
                "UseFIPS": false,
                "UseDualStack": false
              }

            },
            {
              "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.ap-south-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "ap-south-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.ap-southeast-1.amazonaws.com"
```

```
            }
          },
          "params": {
            "Region": "ap-southeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-3.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ca-central-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ca-central-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
```

```
        },
        {
          "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-central-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-central-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-north-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "eu-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.eu-west-1.amazonaws.com"
            }
```

```
      },
      "params": {
        "Region": "eu-west-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
      "expect": {

        "endpoint": {
          "url": "https://license-manager.eu-west-2.amazonaws.com"
        }
      },
      "params": {
        "Region": "eu-west-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.eu-west-3.amazonaws.com"
        }
      },
      "params": {
        "Region": "eu-west-3",
        "UseFIPS": false,

"UseDualStack": false
      }
    },
    {
      "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.me-south-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "me-south-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
```

```
            {
                "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.sa-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "sa-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }

        },
            {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-fips.us-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.us-east-2.amazonaws.com"
                    }
```

```
    },
    "params": {
      "Region": "us-east-2",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-fips.us-east-2.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-2",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region
us-west-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager.us-west-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-west-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-fips.us-west-1.amazonaws.com"
      }
    },
    "params": {

      "Region": "us-west-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
```

```
    {
      "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-west-2.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-west-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

"expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-west-2.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-west-2",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,

        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-east-1.api.aws"
        }
      },
```

```
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-northwest-1",
            "UseFIPS": false,
            "UseDualStack": false
          }

        }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
```

```
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },


      {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.us-gov-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-west-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-west-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
```

```
          "endpoint": {
            "url": "https://license-manager.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
```

```
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "error": "DualStack is enabled
but this partition does not support DualStack"
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
      }
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },

  {
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    },

"params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
```

```
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "error": "DualStack is enabled but this partition does not support DualStack"
              },
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

      "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,

      "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
```

```
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "Missing region",
            "expect": {
              "error": "Invalid Configuration: Missing Region"
            }
          }
        ],
        "version": "1.0"
      }
    }
  },
  "com.amazonaws.licensemanager#AcceptGrant": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

    },
    "output": {
      "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
```

```
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"

    }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#AcceptGrantRequest": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#AcceptGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant
ARN.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#AccessDeniedException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {

"code": "ServiceAccessDenied",
          "httpResponseCode": 401
        },
        "smithy.api#documentation": "<p>Access to resource denied.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 401
      }
    },
    "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
      "type": "enum",
      "members": {
        "DISTRIBUTED_GRANTS_ONLY": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
          }
        },
        "ALL_GRANTS_PERMITTED_BY_ISSUER": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
          }
        }
```

```
        }
      },
      "com.amazonaws.licensemanager#AllowedOperation": {

    "type": "enum",
      "members": {
        "CREATE_GRANT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateGrant"
          }
        },
        "CHECKOUT_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutLicense"
          }
        },
        "CHECKOUT_BORROW_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckoutBorrowLicense"
          }
        },
        "CHECK_IN_LICENSE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CheckInLicense"
          }
        },
        "EXTEND_CONSUMPTION_LICENSE":
{
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ExtendConsumptionLicense"
          }
        },
        "LIST_PURCHASED_LICENSES": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ListPurchasedLicenses"
          }
        },
        "CREATE_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CreateToken"
          }
```

```
          }
        }
      },
      "com.amazonaws.licensemanager#AllowedOperationList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#AllowedOperation"
        },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max":
7
        }
      }
    },
    "com.amazonaws.licensemanager#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 2048
        },
        "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-
z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
      }
    },
    "com.amazonaws.licensemanager#ArnList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Arn"
      }
    },
    "com.amazonaws.licensemanager#AuthorizationException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError":
{
          "code": "AuthorizationFailure",
          "httpResponseCode": 403
        },
        "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n      policy associated with this account.</p>",
```

```
          "smithy.api#error": "client",
          "smithy.api#httpError": 403
        }
      },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
      "type": "structure",
      "members": {
        "LastRunTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes automated discovery.</p>"
      }
    },

  "com.amazonaws.licensemanager#Boolean": {
      "type": "boolean",
      "traits": {
        "smithy.api#default": false
      }
    },
    "com.amazonaws.licensemanager#BorrowConfiguration": {
      "type": "structure",
      "members": {
        "AllowEarlyCheckIn": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxTimeToLiveInMinutes": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
            "smithy.api#required": {}
          }
        }

},
      "traits": {
        "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
      }
    },
```

```
"com.amazonaws.licensemanager#BoxBoolean": {
    "type": "boolean"
},
"com.amazonaws.licensemanager#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licensemanager#BoxLong": {
    "type": "long"
},
"com.amazonaws.licensemanager#CheckInLicense": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"

        },
        {
            "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
    }
```

```
        },
        "com.amazonaws.licensemanager#CheckInLicenseRequest":
{
            "type": "structure",
            "members": {
                "LicenseConsumptionToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License consumption token.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Beneficiary": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License beneficiary.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#CheckInLicenseResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
      },
      {
        "target":
 "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
    }
  },
  "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
    "type": "structure",
    "members":
 {
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
          "smithy.api#required": {}
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementDataList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
          "smithy.api#required": {}
```

```
              }
            },
            "DigitalSignatureMethod": {
              "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
              "traits": {
                "smithy.api#documentation": "<p>Digital signature method. The possible
value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
                "smithy.api#required": {}
              }
            },
            "NodeId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Node ID.</p>"
              }
            },
            "CheckoutMetadata": {
              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                "smithy.api#documentation": "<p>Information about constraints.</p>"
              }
            },
            "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientToken",
              "traits": {
                "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
              }
            },
            "LicenseConsumptionToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
```

```
                  "smithy.api#documentation": "<p>License consumption token.</p>"
              }
          },
          "EntitlementsAllowed":
{
              "target": "com.amazonaws.licensemanager#EntitlementDataList",
              "traits": {
                  "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
              }
          },
          "NodeId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                  "smithy.api#documentation": "<p>Node ID.</p>"
              }
          },
          "SignedToken": {
              "target": "com.amazonaws.licensemanager#SignedToken",
              "traits": {
                  "smithy.api#documentation": "<p>Signed token.</p>"
              }
          },
          "IssuedAt": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                  "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
              }
          },
          "Expiration": {
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                  "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
              }
          },
          "CheckoutMetadata": {
              "target": "com.amazonaws.licensemanager#MetadataList",
              "traits": {
                  "smithy.api#documentation": "<p>Information about constraints.</p>"
              }
          }
      },
      "traits": {
          "smithy.api#output": {}
      }
  },
  "com.amazonaws.licensemanager#CheckoutLicense": {
      "type": "operation",
```

```
"input": {
  "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
},
"output": {
  "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanager#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanager#AuthorizationException"
  },
  {
    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
  },
  {
    "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
  },
  {
    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
  },
  {
    "target": "com.amazonaws.licensemanager#RedirectException"
  },
  {
    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
  },
  {
    "target":
"com.amazonaws.licensemanager#ServerInternalException"
  },
  {
    "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
  },
  {
    "target": "com.amazonaws.licensemanager#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n          <p>If the
account that created the license is the same that is performing the check out, you must\n          specify the account
as the beneficiary.</p>\n        </note>"
}
},
"com.amazonaws.licensemanager#CheckoutLicenseRequest": {
  "type": "structure",
  "members": {
```

          "ProductSKU": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product
SKU.</p>",
              "smithy.api#required": {}
            }
          },
          "CheckoutType": {
            "target": "com.amazonaws.licensemanager#CheckoutType",
            "traits": {
              "smithy.api#documentation": "<p>Checkout type.</p>",
              "smithy.api#required": {}
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"

```
            }
          }
        },
        "traits": {

      "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
        "type": "structure",
        "members": {
          "CheckoutType": {
            "target": "com.amazonaws.licensemanager#CheckoutType",
            "traits": {
              "smithy.api#documentation": "<p>Checkout type.</p>"
            }
          },
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>"
            }
          },
          "EntitlementsAllowed": {
            "target": "com.amazonaws.licensemanager#EntitlementDataList",
            "traits": {
              "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
            }
          },
          "SignedToken": {
            "target": "com.amazonaws.licensemanager#SignedToken",
            "traits": {
              "smithy.api#documentation": "<p>Signed token.</p>"
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          },
          "IssuedAt": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
            }
          },
          "Expiration": {
```

```
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CheckoutType": {
      "type": "enum",
      "members": {
        "PROVISIONAL": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PROVISIONAL"
          }
        },
        "PERPETUAL": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PERPETUAL"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ClientRequestToken": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 36
        }
      }
    },
    "com.amazonaws.licensemanager#ClientToken": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
```

```
          "max": 2048
        },
        "smithy.api#pattern": "^\\S+$"
      }
    },
    "com.amazonaws.licensemanager#ConflictException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "ConflictException",
          "httpResponseCode":
409
        },
        "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 409
      }
    },
    "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
      "type": "structure",
      "members": {
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      },

  "traits": {
        "smithy.api#documentation": "<p>Details about license consumption.</p>"
      }
    },
    "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
      "type": "list",
      "member": {
```

```
                "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
            }
        },
        "com.amazonaws.licensemanager#ConsumptionConfiguration": {
            "type": "structure",
            "members": {
                "RenewType": {
                    "target": "com.amazonaws.licensemanager#RenewType",
                    "traits": {
                        "smithy.api#documentation": "<p>Renewal frequency.</p>"
                    }
                },
                "ProvisionalConfiguration": {
                    "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
                    }
                },
                "BorrowConfiguration": {
                    "target": "com.amazonaws.licensemanager#BorrowConfiguration",
                    "traits": {
                        "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#CreateGrant": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CreateGrantRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CreateGrantResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target":
"com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
```

```
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrantRequest": {
      "type": "structure",
      "members": {
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant name.</p>",
            "smithy.api#required": {}

  }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Principals": {
```

```
                        "target": "com.amazonaws.licensemanager#PrincipalArnList",
                        "traits": {
                            "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n          (ARN):</p>\n          <ul>\n          <li>\n          <p>An Amazon Web Services
account, which includes only the account specified.</p>\n          </li>\n          </ul>\n          <ul>\n          <li>\n
    <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n          </li>\n          </ul>\n
<ul>\n
        <li>\n          <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n          </ul>",
                            "smithy.api#required": {}
                        }
                    },
                    "HomeRegion": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                            "smithy.api#required": {}
                        }
                    },
                    "AllowedOperations": {
                        "target": "com.amazonaws.licensemanager#AllowedOperationList",
                        "traits": {
                            "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
                            "smithy.api#required": {}
                        }
                    }
                },
                "traits": {
                    "smithy.api#input": {}
                }
            },

            "com.amazonaws.licensemanager#CreateGrantResponse": {
                "type": "structure",
                "members": {
                    "GrantArn": {
                        "target": "com.amazonaws.licensemanager#Arn",
                        "traits": {
                            "smithy.api#documentation": "<p>Grant ARN.</p>"
                        }
                    },
                    "Status": {
                        "target": "com.amazonaws.licensemanager#GrantStatus",
                        "traits": {
                            "smithy.api#documentation": "<p>Grant status.</p>"
                        }
                    },
                    "Version": {
```

```
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateGrantVersion":
  {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
           },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
      },
```

```
    "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
        "type": "structure",
        "members": {
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
                }
            },
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant name.</p>"
                }
            },
            "AllowedOperations": {
                "target": "com.amazonaws.licensemanager#AllowedOperationList",
                "traits": {
                    "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>"
                }
            },
            "StatusReason": {
                "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status reason.</p>"
                }
            },
            "SourceVersion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Current version of the grant.</p>"
                }
```

```
                },
                "Options": {
                    "target": "com.amazonaws.licensemanager#Options",
                    "traits": {
                        "smithy.api#documentation": "<p>The options specified for the grant.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
            "type": "structure",
            "members": {
                "GrantArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant ARN.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>New version of the grant.</p>"
                    }
                }
            },
            "traits":
{
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CreateLicense": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
            },
            "errors": [
```

```
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
           },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
       ],
       "traits": {
          "smithy.api#documentation": "<p>Creates a license.</p>"
       }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
       "type": "operation",
       "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
       },
       "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
       },
       "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
           },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a license configuration.</p>\n          <p>A license configuration
is an abstraction of a customer license agreement that can be\n          consumed and enforced by License Manager.
Components include specifications for the license\n          type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n          Dedicated Instance, Dedicated Host, or
 all of these), license affinity  to host (how long a \n          license must be associated with a host), and the number of
licenses purchased and used.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the license configuration.</p>"
            }
          },
          "LicenseCountingType": {
            "target": "com.amazonaws.licensemanager#LicenseCountingType",
             "traits": {
              "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
            }
          },
          "LicenseCountHardLimit": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
```

Exceeding a hard limit\n   blocks the launch of new instances.</p>"
                }
            },
            "LicenseRules": {
                "target":
 "com.amazonaws.licensemanager#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n   vary by dimension, as follows.</p>\n
\n   \n   <p>\n   <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
        <code>licenseAffinityToHost</code> |\n   <code>maximumCores</code> |
<code>minimumCores</code>\n   </p>\n   \n   \n   <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n   <code>maximumCores</code> |
<code>minimumCores</code> |\n   <code>maximumSockets</code> | <code>minimumSockets</code> |\n
        <code>maximumVcpus</code> | <code>minimumVcpus</code>\n   </p>\n   \n
\n
  <p>\n   <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n   <code>maximumSockets</code> |
<code>minimumSockets</code>\n   </p>\n   \n   \n   <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n   <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n   </p>\n   \n   \n   <p>The unit for
<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for
<code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n
<code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n
<code>True</code> and <code>False</code>.</p>"
                }
            },

    "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                    "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
                }
            },
            "DisassociateWhenNotFound": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            },
            "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                    "smithy.api#documentation": "<p>Product information.</p>"
                }
            }

```
          },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
```

```
      "traits": {
        "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {

          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
            "smithy.api#required": {}
          }
        },
        "SourceLicenseContext": {
          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n        For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
            "smithy.api#required": {}
          }
        },
        "DestinationLicenseContext":
 {

          "target": "com.amazonaws.licensemanager#LicenseConversionContext",
          "traits": {
            "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
      "type": "structure",
      "members": {
        "LicenseConversionTaskId": {
          "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
```

eso

```
      "traits": {
              "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a report generator.</p>"
    }
  },
  "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
```

```
    "type": "structure",
    "members": {
      "ReportGeneratorName":
{
        "target": "com.amazonaws.licensemanager#ReportGeneratorName",
        "traits": {
          "smithy.api#documentation": "<p>Name of the report generator.</p>",
          "smithy.api#required": {}
        }
      },
      "Type": {
        "target": "com.amazonaws.licensemanager#ReportTypeList",
        "traits": {
          "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n        <ul>\n        <li>\n          <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n        </li>\n        <li>\n          <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n        </li>\n
</ul>",
          "smithy.api#required": {}

      }
      },
      "ReportContext": {
        "target": "com.amazonaws.licensemanager#ReportContext",
        "traits": {
          "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
          "smithy.api#required": {}
        }
      },
      "ReportFrequency": {
        "target": "com.amazonaws.licensemanager#ReportFrequency",
        "traits": {
          "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
          "smithy.api#required": {}
        }
      },
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientRequestToken",
        "traits": {
          "smithy.api#documentation": "<p>Unique,
case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
          "smithy.api#required": {}
        }
      },
      "Description": {
        "target": "com.amazonaws.licensemanager#String",
```

```
                    "traits": {
                      "smithy.api#documentation": "<p>Description of the report generator.</p>"
                    }
                  },
                  "Tags": {
                    "target": "com.amazonaws.licensemanager#TagList",
                    "traits": {
                      "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
                    }
                  }
                },
                "traits": {
                  "smithy.api#input": {}
                }
              },
              "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
                "type": "structure",
                "members": {
                  "LicenseManagerReportGeneratorArn":
{
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
                    }
                  }
                },
                "traits": {
                  "smithy.api#output": {}
                }
              },
              "com.amazonaws.licensemanager#CreateLicenseRequest": {
                "type": "structure",
                "members": {
                  "LicenseName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                      "smithy.api#documentation": "<p>License name.</p>",
                      "smithy.api#required": {}
                    }
                  },
                  "ProductName": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                      "smithy.api#documentation": "<p>Product
name.</p>",
                      "smithy.api#required": {}
                    }
```

```
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>",
            "smithy.api#required": {}
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#Issuer",
          "traits": {
            "smithy.api#documentation": "<p>License issuer.</p>",
            "smithy.api#required": {}
          }
        },
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region for the license.</p>",
            "smithy.api#required": {}

  }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
            "smithy.api#required": {}
          }
        },
        "Entitlements": {
          "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>",
            "smithy.api#required": {}
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>",
            "smithy.api#required": {}
          }
        },
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
```

"smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional configuration for workloads\n   running with continuous connectivity. Choose a borrow configuration for workloads with offline\n   usage.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license.</p>"
                }
            },
            "ClientToken": {
                "target": "com.amazonaws.licensemanager#ClientToken",
                "traits": {
                    "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseResponse": {
        "type": "structure",
        "members": {
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#LicenseStatus",
                "traits": {
                    "smithy.api#documentation": "<p>License status.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License version.</p>"
                }
            }
        },
        "traits": {

```
            "smithy.api#output": { }
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersion": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#RedirectException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
        }
    },
    "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
        "type": "structure",
        "members": {
            "LicenseArn": {
                "target":
"com.amazonaws.licensemanager#Arn",
                "traits": {
```

              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License name.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Product name.</p>",
              "smithy.api#required": {}
            }
          },
          "Issuer": {
            "target": "com.amazonaws.licensemanager#Issuer",
            "traits":

{

              "smithy.api#documentation": "<p>License issuer.</p>",
              "smithy.api#required": {}
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
               "smithy.api#documentation": "<p>Information about the license.</p>"
            }
          },
          "Entitlements": {

```
              "target": "com.amazonaws.licensemanager#EntitlementList",
              "traits": {
                 "smithy.api#documentation": "<p>License entitlements.</p>",
                 "smithy.api#required": {}
              }
           },
           "ConsumptionConfiguration": {
              "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
              "traits": {
                 "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n       running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n       usage.</p>",
                 "smithy.api#required": {}
              }
           },

    "Status": {
              "target": "com.amazonaws.licensemanager#LicenseStatus",
              "traits": {
                 "smithy.api#documentation": "<p>License status.</p>",
                 "smithy.api#required": {}
              }
           },
           "ClientToken": {
              "target": "com.amazonaws.licensemanager#ClientToken",
              "traits": {
                 "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                 "smithy.api#required": {}
              }
           },
           "SourceVersion": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                 "smithy.api#documentation": "<p>Current version of the license.</p>"
              }
           }
        },
        "traits": {

 "smithy.api#input": {}
        }
     },
     "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
           "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
```

```
              "traits": {
                 "smithy.api#documentation": "<p>License ARN.</p>"
              }
           },
           "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                 "smithy.api#documentation": "<p>New version of the license.</p>"
              }
           },
           "Status": {
              "target": "com.amazonaws.licensemanager#LicenseStatus",
              "traits": {
                 "smithy.api#documentation": "<p>License status.</p>"
              }
           }
        },
        "traits": {
           "smithy.api#output":
{}
        }
     },
     "com.amazonaws.licensemanager#CreateToken": {
        "type": "operation",
        "input": {
           "target": "com.amazonaws.licensemanager#CreateTokenRequest"
        },
        "output": {
           "target": "com.amazonaws.licensemanager#CreateTokenResponse"
        },
        "errors": [
           {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
           },
           {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
           },
           {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
           },
           {
              "target": "com.amazonaws.licensemanager#RedirectException"
           },
           {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
           },
```

```
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
            }
        },
        "com.amazonaws.licensemanager#CreateTokenRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits":
 {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
                        "smithy.api#required": {}
                    }
                },
                "RoleArns": {
                    "target": "com.amazonaws.licensemanager#ArnList",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n        License Manager does not check whether the roles are in use.</p>"
                    }
                },
                "ExpirationInDays": {
                    "target": "com.amazonaws.licensemanager#Integer",
                    "traits": {
                        "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
                    }
                },
                "TokenProperties": {
                    "target": "com.amazonaws.licensemanager#MaxSize3StringList",
                    "traits": {
                        "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n        to the amr claim of the JWT token.</p>"
                    }
```

```
        },
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateTokenResponse": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",

          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
        "TokenType": {
          "target": "com.amazonaws.licensemanager#TokenType",
          "traits": {
            "smithy.api#documentation": "<p>Token type.</p>"
          }
        },
        "Token": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DateTime": {
      "type": "timestamp"
    },
    "com.amazonaws.licensemanager#DatetimeRange": {
      "type": "structure",
      "members": {
        "Begin": {
```

```
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Start of the time range.</p>",
            "smithy.api#required": {}
          }
        },
        "End": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>End of the time range.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
        }
    },
    "com.amazonaws.licensemanager#DeleteGrantRequest": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "StatusReason": {
                "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                "traits": {
                    "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Current version of the
grant.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteGrantResponse": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>"
                }
```

          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
               "smithy.api#documentation":
"<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"

},
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],

```
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
      },

    "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
      "type": "structure",
      "members":
{
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the license configuration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
```

```
          "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n          <p>This action
deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It
has no effect on the previous reports from this generator.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

 "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the license.</p>",
              "smithy.api#required":
{}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

```
        },
        "com.amazonaws.licensemanager#DeleteLicenseResponse": {
            "type": "structure",
            "members": {
                "Status": {
                    "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>License status.</p>"
                    }
                },
                "DeletionDate": {
                    "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteToken": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RedirectException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
```

```
        {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
        }
    },
    "com.amazonaws.licensemanager#DeleteTokenRequest": {
        "type": "structure",
        "members": {
            "TokenId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token ID.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteTokenResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}

    }
    },
    "com.amazonaws.licensemanager#DigitalSignatureMethod": {
        "type": "enum",
        "members": {
            "JWT_PS384": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "JWT_PS384"
                }
            }
        }
    },
    "com.amazonaws.licensemanager#Entitlement": {
        "type": "structure",
        "members": {
            "Name": {
```

```
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement name.</p>",
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#Long",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
          }
        },
        "Overage": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement unit.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowCheckIn": {
          "target":
"com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementData": {
      "type": "structure",
      "members": {
        "Name": {
```

                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Entitlement data name.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Value": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Entitlement
data value.</p>"
                    }
                },
                "Unit": {
                    "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
                    "traits": {
                        "smithy.api#documentation": "<p>Entitlement data unit.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
            }
        },
        "com.amazonaws.licensemanager#EntitlementDataList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanager#EntitlementData"
            }
        },
        "com.amazonaws.licensemanager#EntitlementDataUnit": {
            "type": "enum",
            "members": {
                "COUNT": {
                    "target": "smithy.api#Unit",
                    "traits": {

"smithy.api#enumValue": "Count"
                    }
                },
                "NONE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "None"
                    }
                },
                "SECONDS": {

```
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Seconds"
        }
      },
      "MICROSECONDS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Microseconds"
        }
      },
      "MILLISECONDS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Milliseconds"
        }
      },
      "BYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes"
        }
      },
      "KILOBYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes"
        }
      },
      "MEGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabytes"
        }
      },
      "GIGABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabytes"
        }
      },
      "TERABYTES": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabytes"
        }
      },
      "BITS":
```

```
{
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits"
        }
      },
      "KILOBITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits"
        }
      },
      "MEGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabits"
        }
      },
      "GIGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits"
        }
      },
      "TERABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits"

}
      },
      "PERCENT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Percent"
        }
      },
      "BYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes/Second"
        }
      },
      "KILOBYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes/Second"
        }
```

          },
          "MEGABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Megabytes/Second"
            }
          },
          "GIGABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
             "traits": {
              "smithy.api#enumValue": "Gigabytes/Second"
            }
          },
          "TERABYTES_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Terabytes/Second"
            }
          },
          "BITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Bits/Second"
            }
          },
          "KILOBITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Kilobits/Second"
            }
          },
          "MEGABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue":
"Megabits/Second"
            }
          },
          "GIGABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Gigabits/Second"
            }
          },
          "TERABITS_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Terabits/Second"

```json
        }
      },
      "COUNT_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Count/Second"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#EntitlementList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#Entitlement"
    }
  },
  "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#EntitlementUnit": {
    "type": "enum",
    "members": {
      "COUNT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Count"
        }
      },
      "NONE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "None"
        }
      },
      "SECONDS":
{
        "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue": "Seconds"
      }
    },
    "MICROSECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Microseconds"
      }
    },
    "MILLISECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Milliseconds"
      }
    },
    "BYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS": {
      "target": "smithy.api#Unit",
```

```
          "traits": {
            "smithy.api#enumValue": "Bits"
          }
        },
        "KILOBITS": {
          "target": "smithy.api#Unit",
          "traits": {


            "smithy.api#enumValue": "Kilobits"
          }
        },
        "MEGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits"
          }
        },
        "GIGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits"
          }
        },
        "TERABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits"
          }
        },
        "PERCENT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Percent"
          }
        },
        "BYTES_PER_SECOND": {
          "target":
"smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bytes/Second"
          }
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
```

```
"MEGABYTES_PER_SECOND": {
    "target": "smithy.api#Unit",
    "traits": {
        "smithy.api#enumValue": "Megabytes/Second"
    }
},
"GIGABYTES_PER_SECOND": {
    "target": "smithy.api#Unit",
    "traits": {
        "smithy.api#enumValue": "Gigabytes/Second"
    }
},
"TERABYTES_PER_SECOND": {
    "target": "smithy.api#Unit",
    "traits": {

"smithy.api#enumValue": "Terabytes/Second"
    }
},
"BITS_PER_SECOND": {
    "target": "smithy.api#Unit",
    "traits": {
        "smithy.api#enumValue": "Bits/Second"
    }
},
"KILOBITS_PER_SECOND": {
    "target": "smithy.api#Unit",
    "traits": {
        "smithy.api#enumValue": "Kilobits/Second"
    }
},
"MEGABITS_PER_SECOND": {
    "target": "smithy.api#Unit",
    "traits": {
        "smithy.api#enumValue": "Megabits/Second"
    }
},
"GIGABITS_PER_SECOND": {
    "target": "smithy.api#Unit",
    "traits": {
        "smithy.api#enumValue": "Gigabits/Second"
    }
},

"TERABITS_PER_SECOND": {
    "target": "smithy.api#Unit",
    "traits": {
        "smithy.api#enumValue": "Terabits/Second"
```

```
            }
          },
          "COUNT_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count/Second"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsage": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage name.</p>",
            "smithy.api#required": {}
          }
        },
        "ConsumedValue": {
          "target": "com.amazonaws.licensemanager#String",

          "traits": {
            "smithy.api#documentation": "<p>Resource usage consumed.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList":
```

```
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementUsage"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>",
```

```
                    "smithy.api#required": {}

                }
            },
            "DryRun": {
                "target": "com.amazonaws.licensemanager#Boolean",
                "traits": {
                    "smithy.api#default": false,
                    "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
        "type": "structure",
        "members": {
            "LicenseConsumptionToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>License consumption token.</p>"
                }
            },

            "Expiration": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#FailedDependencyException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            },
            "ErrorCode": {
                "target": "com.amazonaws.licensemanager#String"
            }
        },
```

```
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "FailedDependency",
            "httpResponseCode": 424
          },
          "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 424
        }
      },
      "com.amazonaws.licensemanager#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#FilterName",
            "traits": {
              "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
            }
          },
          "Values": {
            "target": "com.amazonaws.licensemanager#FilterValues",
            "traits": {
              "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n        describe operation. Filters can be used
to match a set of resources by specific criteria,\n        such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanager#FilterLimitExceededException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "FilterLimitExceeded",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
          "smithy.api#error": "client",
```

```
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Filter"
        }
      },
      "com.amazonaws.licensemanager#FilterName": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#FilterValue": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#FilterValues": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#FilterValue",
          "traits": {
            "smithy.api#xmlName": "item"
          }
        }
      },
      "com.amazonaws.licensemanager#Filters": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Filter",
          "traits": {
            "smithy.api#xmlName": "item"
          }
        }
      },
      "com.amazonaws.licensemanager#GetAccessToken": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
```

```
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
 tokens\n        are valid for one hour.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetAccessTokenRequest": {
            "type": "structure",
            "members": {
                "Token": {
                    "target": "com.amazonaws.licensemanager#TokenString",
                    "traits": {
                        "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
                        "smithy.api#required": {}
                    }
                },
                "TokenProperties": {
                    "target": "com.amazonaws.licensemanager#MaxSize3StringList",
                    "traits": {
                        "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
            "type": "structure",
            "members": {
                "AccessToken": {
                    "target": "com.amazonaws.licensemanager#TokenString",
                    "traits": {
                        "smithy.api#documentation": "<p>Temporary access token.</p>"
                    }
                }
```

```
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetGrant": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetGrantRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetGrantResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"

          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetGrantRequest": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
```

```json
            "smithy.api#required": {}
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrantResponse": {
      "type": "structure",
      "members": {
        "Grant": {
          "target": "com.amazonaws.licensemanager#Grant",
          "traits": {
            "smithy.api#documentation": "<p>Grant details.</p>"
          }
        }
      },
      "traits": {

"smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
              },
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"

        },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetLicenseConfiguration": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
             },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
          }
        },
        "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
```

```
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "traits":
{
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
        "type": "structure",
        "members": {
          "LicenseConfigurationId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
            }
          },
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
            }
          },
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the license configuration.</p>"
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the license configuration.</p>"
            }
          },
          "LicenseCountingType": {
            "target": "com.amazonaws.licensemanager#LicenseCountingType",
            "traits": {
              "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
            }
          },
          "LicenseRules": {
            "target": "com.amazonaws.licensemanager#StringList",
```

```
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",


        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License configuration status.</p>"
        }


      },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
        }
      },
      "ConsumedLicenseSummaryList": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
        }
      },
      "ManagedResourceSummaryList": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
        }
      },
```

```
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
           "traits": {
             "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
           }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
            "smithy.api#documentation": "<p>Product information.</p>"
          }
        },
        "AutomatedDiscoveryInformation": {
          "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
          "traits": {
            "smithy.api#documentation": "<p>Automated discovery information.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTask": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
```

```
                        {

   "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                    },
                    {
                        "target": "com.amazonaws.licensemanager#ServerInternalException"
                    }
                ],
                "traits": {
                    "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
                }
            },
            "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
                "type": "structure",
                "members": {
                    "LicenseConversionTaskId": {
                        "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                        "traits": {
                            "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
                            "smithy.api#required": {}
                        }
                    }
                },
                "traits": {
                    "smithy.api#input": {}
                }
            },
            "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
                "type": "structure",
                "members": {
                    "LicenseConversionTaskId": {
                        "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                        "traits": {
                            "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
                        }
                    },
                    "ResourceArn": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
                        }
                    },
                    "SourceLicenseContext": {
                        "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                        "traits": {
                            "smithy.api#documentation": "<p>Information
```

about the license type converted from.</p>"
                }
            },
            "DestinationLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
                }
            },

    "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                }
            },
            "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                }
            },
            "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":

```
{
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

        },
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
    }
},
"com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
    "type": "structure",
    "members": {
        "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
                "smithy.api#required":
```

```
{}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
          "type": "structure",
          "members": {
            "ReportGenerator": {
              "target": "com.amazonaws.licensemanager#ReportGenerator",
              "traits": {
                "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#GetLicenseRequest": {
          "type": "structure",
          "members": {
            "LicenseArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                "smithy.api#required": {}
              }
            },
            "Version": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>License version.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#GetLicenseResponse": {
          "type": "structure",
          "members": {
```

```
      "License": {
        "target": "com.amazonaws.licensemanager#License",
        "traits": {
          "smithy.api#documentation": "<p>License details.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},

"com.amazonaws.licensemanager#GetLicenseUsage": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {

"target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
    "type": "structure",
    "members": {
```

```
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
        "type": "structure",
        "members": {
          "LicenseUsage": {
            "target":
"com.amazonaws.licensemanager#LicenseUsage",
            "traits": {
              "smithy.api#documentation": "<p>License usage details.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
```

```
          ],
        "traits": {
          "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "S3BucketArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"

  }
        },
        "SnsTopicArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
          }
        },
        "OrganizationConfiguration": {
          "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n        cross-account discovery.</p>"
          }
        },
        "EnableCrossAccountsDiscovery": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
          }

      },
        "LicenseManagerResourceShareArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
```

```
Manager management account \n        provides member accounts with access to this share.</p>"
                }
              }
            },
          "traits": {
            "smithy.api#output": {}
          }
        }
      },
      "com.amazonaws.licensemanager#Grant": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
              "smithy.api#required": {}
            }
          },
          "GrantName": {

            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant name.</p>",
              "smithy.api#required": {}
            }
          },
          "ParentArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Parent ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>License ARN.</p>",
              "smithy.api#required": {}
            }
          },
          "GranteePrincipalArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {

              "smithy.api#documentation": "<p>The grantee principal ARN.</p>",
              "smithy.api#required": {}
            }
          },
```

```
        "HomeRegion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Home Region of the grant.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantStatus": {
          "target": "com.amazonaws.licensemanager#GrantStatus",
          "traits": {
            "smithy.api#documentation": "<p>Grant status.</p>",
            "smithy.api#required": {}
          }
        },
        "StatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Grant status reason.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant version.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
          "traits": {
            "smithy.api#documentation": "<p>Granted operations.</p>",
            "smithy.api#required": {}
          }
        },
        "Options": {
          "target": "com.amazonaws.licensemanager#Options",
          "traits": {
            "smithy.api#documentation": "<p>The options specified for the grant.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Describes a grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#GrantList": {
      "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanager#Grant"
        }
    },
    "com.amazonaws.licensemanager#GrantStatus": {
      "type": "enum",
      "members": {
        "PENDING_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_WORKFLOW"
          }
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_ACCEPT"
          }
        },
        "REJECTED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"


        }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
```

```
            }
          },
          "DISABLED": {
            "target": "smithy.api#Unit",
            "traits":
{
              "smithy.api#enumValue": "DISABLED"
            }
          },
          "WORKFLOW_COMPLETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "WORKFLOW_COMPLETED"
            }
          }
        }
    },
    "com.amazonaws.licensemanager#GrantedLicense": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>"
          }

    },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#IssuerDetails",
          "traits": {
```

```
              "smithy.api#documentation": "<p>Granted license issuer.</p>"
            }
          },
          "HomeRegion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Home Region of the granted license.</p>"

          }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>Granted license status.</p>"
            }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
ISO8601-UTC format.</p>"
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
            }
          },
          "Entitlements": {
            "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>"
            }
          },
          "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Granted license metadata.</p>"
            }
          },
          "CreateTime": {
```

```
              "target": "com.amazonaws.licensemanager#ISO8601DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

    }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Version of the granted license.</p>"
            }
          },
          "ReceivedMetadata": {
            "target": "com.amazonaws.licensemanager#ReceivedMetadata",
            "traits": {
              "smithy.api#documentation": "<p>Granted license received metadata.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
        }
      },
      "com.amazonaws.licensemanager#GrantedLicenseList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#GrantedLicense"
        }
      },
      "com.amazonaws.licensemanager#ISO8601DateTime":
{
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
        }
      },
      "com.amazonaws.licensemanager#Integer": {
        "type": "integer"
      },
      "com.amazonaws.licensemanager#InvalidParameterValueException": {
        "type": "structure",
        "members": {
          "Message": {
```

```
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidParameterValueProvided",
            "httpResponseCode": 400
          },
          "smithy.api#documentation":
  "<p>One or more parameter values are not valid.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#InvalidResourceStateException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidResourceState",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n          <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },

      "com.amazonaws.licensemanager#InventoryFilter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the filter.</p>",
              "smithy.api#required": {}
            }
          },
          "Condition": {
            "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
            "traits": {
              "smithy.api#documentation": "<p>Condition of the filter.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An inventory filter.</p>"
    }
  },
  "com.amazonaws.licensemanager#InventoryFilterCondition": {
    "type": "enum",
    "members": {
      "EQUALS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "EQUALS"
        }
      },
      "NOT_EQUALS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "NOT_EQUALS"
        }
      },
      "BEGINS_WITH": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "BEGINS_WITH"
        }
      },
      "CONTAINS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "CONTAINS"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#InventoryFilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#InventoryFilter"
```

```
        }
      },
      "com.amazonaws.licensemanager#Issuer": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>",
              "smithy.api#required": {}
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
 of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#IssuerDetails": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>"
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n        and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
            }
          }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
        }
    },
    "com.amazonaws.licensemanager#License": {
        "type": "structure",
        "members": {
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
                }
            },
            "LicenseName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {

            "smithy.api#documentation": "<p>License name.</p>"
                }
            },
            "ProductName": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Product name.</p>"
                }
            },
            "ProductSKU": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Product SKU.</p>"
                }
            },
            "Issuer": {
                "target": "com.amazonaws.licensemanager#IssuerDetails",
                "traits": {
                    "smithy.api#documentation": "<p>License issuer.</p>"
                }
            },
            "HomeRegion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {

            "smithy.api#documentation": "<p>Home Region of the license.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#LicenseStatus",
```

```
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
          "Validity": {
            "target": "com.amazonaws.licensemanager#DatetimeRange",
            "traits": {
              "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          },
          "Entitlements": {

    "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>"
            }
          },
          "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>License metadata.</p>"
            }
          },
          "CreateTime": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>License
creation time.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
```

```
                    }
                },
                "traits": {
                    "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
                }
            },
            "com.amazonaws.licensemanager#LicenseConfiguration": {
                "type": "structure",
                "members": {
                    "LicenseConfigurationId": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
                        }
                    },
                    "LicenseConfigurationArn": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
                        }
                    },
                    "Name": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Name of the license configuration.</p>"
                        }
                    },
                    "Description": {
                        "target": "com.amazonaws.licensemanager#String",
                        "traits": {
                            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
                        }
                    },
                    "LicenseCountingType": {
                        "target": "com.amazonaws.licensemanager#LicenseCountingType",
                        "traits": {
                            "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
                        }
                    },
                    "LicenseRules": {
                        "target": "com.amazonaws.licensemanager#StringList",
                        "traits": {
                            "smithy.api#documentation": "<p>License rules.</p>"
                        }
                    },
                    "LicenseCount": {
                        "target": "com.amazonaws.licensemanager#BoxLong",
                        "traits": {
```

```
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
          }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the license configuration.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

        }
        },
        "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
          }
        },
        "ManagedResourceSummaryList": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
          }
        },
        "ProductInformationList": {
```

```
            "target": "com.amazonaws.licensemanager#ProductInformationList",
            "traits": {
              "smithy.api#documentation": "<p>Product information.</p>"
            }
          },
          "AutomatedDiscoveryInformation": {

            "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
            "traits": {
              "smithy.api#documentation": "<p>Automated discovery information.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
agreement that can be\n        consumed and enforced by License Manager. Components include specifications for
the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n        must be associated with a
host), and the number of licenses purchased and used.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
        "type": "structure",
        "members": {
          "ResourceArn":
  {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
          "ResourceType": {
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of server resource.</p>"
            }
          },
          "ResourceOwnerId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
consuming licenses.</p>"
            }
          },
          "AssociationTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
```

```
          "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
        }
      },
      "AmiAssociationScope": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
    }
  },
  "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
    "type": "enum",
    "members":
{
      "AVAILABLE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "AVAILABLE"
        }
      },
      "DISABLED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DISABLED"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
```

```
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits":
{
              "smithy.api#documentation": "<p>Type of resource.</p>"
          }
        },
        "ResourceStatus": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the resource.</p>"
          }
        },
        "ResourceOwnerId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
          }
        },
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
          }
        },

    "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurations": {
      "type": "list",
      "member": {
```

```
              "target": "com.amazonaws.licensemanager#LicenseConfiguration"
            }
        },
      "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
        "members": {
          "UsageOperation": {
            "target": "com.amazonaws.licensemanager#UsageOperation",
            "traits": {
              "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n        </p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {

    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
            }
          },
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
            }
          },
          "DestinationLicenseContext": {
            "target":
```

```
        "com.amazonaws.licensemanager#LicenseConversionContext",
              "traits": {
                "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
              "traits": {
                "smithy.api#documentation": "<p>The status of the conversion task.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
              }
            },
            "StartTime": {
              "target": "com.amazonaws.licensemanager#DateTime",
              "traits": {
                "smithy.api#documentation":
  "<p>The time the conversion task was started at.</p>"
              }
            },
            "LicenseConversionTime": {
              "target": "com.amazonaws.licensemanager#DateTime",
              "traits": {
                "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
              }
            },
            "EndTime": {
              "target": "com.amazonaws.licensemanager#DateTime",
              "traits": {
                "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
          }
        },
        "com.amazonaws.licensemanager#LicenseConversionTaskId": {
          "type": "string",
          "traits": {
```

```
      "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
      "type": "enum",
      "members": {
        "IN_PROGRESS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "IN_PROGRESS"
          }
        },
        "SUCCEEDED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SUCCEEDED"
          }
        },
        "FAILED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseConversionTasks":
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConversionTask"
      }
    },
    "com.amazonaws.licensemanager#LicenseCountingType": {
      "type": "enum",
      "members": {
        "VCPU": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "vCPU"
          }
        },
        "INSTANCE": {
          "target": "smithy.api#Unit",
```

```json
      "traits": {
        "smithy.api#enumValue": "Instance"
      }
    },
    "CORE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Core"
      }
    },
    "SOCKET": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Socket"
      }
    }
  }
},
"com.amazonaws.licensemanager#LicenseDeletionStatus": {
  "type": "enum",
  "members": {
    "PENDING_DELETE": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "PENDING_DELETE"
      }
    },
    "DELETED": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "DELETED"
      }
    }
  }
},
"com.amazonaws.licensemanager#LicenseList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanager#License"
  }
},
"com.amazonaws.licensemanager#LicenseOperationFailure": {

  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
```

```
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Resource type.</p>"
        }
      },
      "ErrorMessage": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Error message.</p>"
        }
      },
      "FailureTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {

    "smithy.api#documentation": "<p>Failure time.</p>"
        }
      },
      "OperationName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Name of the operation.</p>"
        }
      },
      "ResourceOwnerId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
        }
      },
      "OperationRequestedBy": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
        }
      },
      "MetadataList": {

        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
          "smithy.api#documentation": "<p>Reserved.</p>"
        }
      }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseOperationFailureList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecification": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

              "smithy.api#required": {}
            }
          },
          "AmiAssociationScope": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseSpecifications": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseSpecification"
        }
      },
      "com.amazonaws.licensemanager#LicenseStatus": {
        "type": "enum",
        "members": {
          "AVAILABLE": {
            "target": "smithy.api#Unit",
```

```
            "traits": {
                "smithy.api#enumValue": "AVAILABLE"
            }
        },
        "PENDING_AVAILABLE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "PENDING_AVAILABLE"
            }
        },
        "DEACTIVATED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "DEACTIVATED"
            }
        },
        "SUSPENDED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "SUSPENDED"
            }
        },
        "EXPIRED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "EXPIRED"
            }
        },

    "PENDING_DELETE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "PENDING_DELETE"
            }
        },
        "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "DELETED"
            }
        }
    }
},
"com.amazonaws.licensemanager#LicenseUsage": {
    "type": "structure",
    "members": {
        "EntitlementUsages": {
            "target": "com.amazonaws.licensemanager#EntitlementUsageList",
```

```
              "traits": {
                "smithy.api#documentation": "<p>License entitlement usages.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

          }
        },
        "com.amazonaws.licensemanager#LicenseUsageException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "LicenseUsageFailure",
              "httpResponseCode": 412
            },
            "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 412
          }
        },
        "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
          },
          "output": {

        "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
            },
            {
```

```
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurationAssociations": {
```

```
                "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
                }
            },
            "NextToken": {

    "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListDistributedGrants": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
```

```
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
    "type": "structure",
    "members": {
      "GrantArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {

        "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n          <ul>\n          <li>\n          <p>\n                    <code>LicenseArn</code>\n          </p>\n
    </li>\n          <li>\n          <p>\n                    <code>GrantStatus</code>\n          </p>\n          </li>\n
  <li>\n          <p>\n                    <code>GranteePrincipalARN</code>\n          </p>\n          </li>\n
<li>\n          <p>\n                    <code>ProductSKU</code>\n          </p>\n          </li>\n          <li>\n
<p>\n                    <code>LicenseIssuerName</code>\n
      </p>\n          </li>\n          </ul>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#MaxSize100",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
    "type": "structure",
    "members": {
      "Grants": {
```

```
            "target": "com.amazonaws.licensemanager#GrantList",

      "traits": {
                  "smithy.api#documentation": "<p>Distributed grant details.</p>"
               }
            },
            "NextToken": {
               "target": "com.amazonaws.licensemanager#String",
               "traits": {
                  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
         },
         "errors": [
            {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
               "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
               "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
               "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
         }
      },
      "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
```

```
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
           "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
      "type": "structure",
      "members": {
        "LicenseOperationFailureList": {
          "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
          "traits": {
            "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurations": {
```

```
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
        },

    "output": {
            "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the license configurations
 for your account.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
        "type": "structure",
        "members": {
            "LicenseConfigurationArns": {
                "target": "com.amazonaws.licensemanager#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
```

```
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next
set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#Filters",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n     are supported:</p>\n     <ul>\n       <li>\n         <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n          Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n          Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n       </li>\n       <li>\n
<p>\n            <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
enforcement is used. \n          Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n
   </li>\n       <li>\n         <p>\n            <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n          Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n       </li>\n     </ul>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
      "type": "structure",
      "members": {
        "LicenseConfigurations": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurations",
          "traits": {
            "smithy.api#documentation": "<p>Information about the license configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",

        "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
        "type": "structure",
        "members": {
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {

"smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "Filters": {
```

```
                "target": "com.amazonaws.licensemanager#Filters",
                "traits": {
                    "smithy.api#documentation": "<p>\n        Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n      </p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
        "type": "structure",
        "members": {
            "LicenseConversionTasks": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
                "traits": {
                    "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target":
"com.amazonaws.licensemanager#AuthorizationException"
            },
```

```
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n        <ul>\n          <li>\n            <p>\n              <code>LicenseConfigurationArn</code>\n
</p>\n          </li>\n        </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits":
{
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
```

```
        },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
      "type": "structure",
      "members": {
        "ReportGenerators": {
          "target": "com.amazonaws.licensemanager#ReportGeneratorList",
          "traits": {
            "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }

      }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

```json
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {

    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {
        "LicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
          }
        },
        "NextToken": {
```

```
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        }
```

```json
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#LicenseList",
          "traits": {
            "smithy.api#documentation": "<p>License details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenses": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicensesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicensesResponse"
      },
      "errors": [
```

```json
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
        }
    },
    "com.amazonaws.licensemanager#ListLicensesRequest": {
        "type": "structure",
        "members": {
            "LicenseArns": {
                "target": "com.amazonaws.licensemanager#ArnList",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
                }
            },

        "Filters": {
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n        <p>\n            <code>Beneficiary</code>\n        </p>\n
    </li>\n        <li>\n        <p>\n            <code>ProductSKU</code>\n        </p>\n        </li>\n
  <li>\n        <p>\n            <code>Fingerprint</code>\n        </p>\n        </li>\n        <li>\n
<p>\n            <code>Status</code>\n        </p>\n        </li>\n        </ul>"
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token
```

for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListLicensesResponse": {
            "type": "structure",
            "members": {
                "Licenses": {
                    "target": "com.amazonaws.licensemanager#LicenseList",
                    "traits": {
                        "smithy.api#documentation": "<p>License details.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {

"smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListReceivedGrants": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {

              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
             },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n   recipient as this Amazon Web Services account, your organization, or an organizational unit\n
   (OU) to which this member account belongs.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },

```
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target":
"com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
              "smithy.api#required": {}
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
    <ul>\n        <li>\n          <p>\n              <code>ParentArn</code>\n          </p>\n        </li>\n
<li>\n          <p>\n              <code>GranteePrincipalArn</code>\n          </p>\n        </li>\n      </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

"com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
    "type": "structure",
    "members": {
      "Grants": {
        "target": "com.amazonaws.licensemanager#GrantList",
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
},
    "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
      "type": "structure",
      "members": {
        "GrantArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>ProductSKU</code>\n   </p>\n
          \n   \n   <p>\n   <code>LicenseIssuerName</code>\n   </p>\n
\n   \n   <p>\n   <code>LicenseArn</code>\n   </p>\n   \n
\n   <p>\n   <code>GrantStatus</code>\n   </p>\n   \n   \n
<p>\n
        <code>GranterAccountId</code>\n   </p>\n   \n   "
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },

```
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
      "type": "structure",
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",
          "traits": {
            "smithy.api#documentation": "<p>Received grant details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicenses": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists received licenses.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
{
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {

      "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {

```
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanager#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n        <p>\n        <code>Beneficiary</code>\n        </p>\n
        </li>\n        <li>\n        <p>\n        <code>ProductSKU</code>\n
        </p>\n        </li>\n        </ul>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#MaxSize100",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
        "type": "structure",
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#GrantedLicenseList",
            "traits": {
              "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },

        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n          <li>\n            <p>\n               <code>ProductSKU</code>\n            </p>\n
   </li>\n          <li>\n            <p>\n               <code>Status</code>\n            </p>\n          </li>\n
<li>\n            <p>\n               <code>Fingerprint</code>\n            </p>\n          </li>\n          <li>\n
<p>\n               <code>IssuerName</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Beneficiary</code>\n            </p>\n          </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
      "type": "structure",
```

```
"members": {
  "Licenses": {
    "target": "com.amazonaws.licensemanager#GrantedLicenseList",
    "traits": {
      "smithy.api#documentation": "<p>Received license details.</p>"
    }
  },

  "NextToken": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
      "smithy.api#documentation": "<p>Token for the next set of results.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanager#ListResourceInventory": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#FailedDependencyException"
    },
    {
      "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#InventoryFilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n    <ul>\n      <li>\n        <p>\n          <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n         Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
      </li>\n      <li>\n        <p>\n          <code>application_name</code> - The name of the
application. \n         Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
   </li>\n      <li>\n        <p>\n          <code>license_included</code> - The type of license included. \n
      Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n         Possible values
are <code>sql-server-enterprise</code> | \n          <code>sql-server-standard</code> | \n          <code>sql-
server-web</code> | \n          <code>windows-server-datacenter</code>.</p>\n      </li>\n      <li>\n
   <p>\n          <code>platform</code> - The platform of the resource.\n         Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n      </li>\n      <li>\n        <p>\n
      <code>resource_id</code> - The ID of the resource. \n         Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n      </li>\n      <li>\n        <p>\n
   <code>tag:<key></code> - The key/value combination of a tag assigned \n          to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n          <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n      </li>\n    </ul>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
```

```
    "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
      "type": "structure",
      "members": {
        "ResourceInventoryList": {
          "target": "com.amazonaws.licensemanager#ResourceInventoryList",
          "traits": {
            "smithy.api#documentation": "<p>Information
about the resources.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

      "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the tags.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListTokens": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListTokensRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListTokensResponse"
        },
        "errors":
[
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists your tokens.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListTokensRequest": {
      "type": "structure",
      "members": {
        "TokenIds": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {

            "smithy.api#documentation": "<p>Token IDs.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n        <li>\n            <p>\n                <code>LicenseArns</code>\n            </p>\n        </li>\n
</ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
```

```
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
          "Tokens": {
            "target": "com.amazonaws.licensemanager#TokenList",
            "traits": {
              "smithy.api#documentation": "<p>Received token details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
```

```
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n        consumption details by resource at a selected point in time. Use this action to audit the\n        current
license consumption for any license inventory and configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
    "type": "structure",
    "members": {
      "LicenseConfigurationArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanager#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#Filters",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n            <li>\n                <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n            </li>\n
<li>\n                <p>\n                    <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n            </li>\n            <li>\n                <p>\n
    <code>resourceAccount</code> - The ID of the account that owns the resource. \n            Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n            </li>\n        </ul>"
        }
      }
```

```
          },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
        "type": "structure",
        "members": {

"LicenseConfigurationUsageList": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#Location": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#Long": {
        "type": "long"
      },
      "com.amazonaws.licensemanager#ManagedResourceSummary": {
        "type": "structure",
        "members": {
          "ResourceType":
{
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
            }
          },
          "AssociationCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
            }
          }
```

      },
      "traits": {
        "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
      }
    },
    "com.amazonaws.licensemanager#MaxSize100": {

      "type": "integer",
      "traits": {
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "com.amazonaws.licensemanager#MaxSize3StringList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 3
        }
      }
    },
    "com.amazonaws.licensemanager#Message": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Metadata": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The key name.</p>"
          }
        },

    "Value": {
          "target": "com.amazonaws.licensemanager#String",

```
            "traits": {
              "smithy.api#documentation": "<p>The value.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
        }
      },
      "com.amazonaws.licensemanager#MetadataList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Metadata"
        }
      },
      "com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#Options": {
        "type": "structure",
        "members": {
          "ActivationOverrideBehavior": {
            "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
            "traits": {
              "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n        Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n        <ul>\n        <li>\n              <p>As a license administrator, you can
optionally specify an\n              <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
</li>\n        <li>\n          <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n            account in your
organization.</p>\n        </li>\n        <li>\n              <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n              <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n          are activating the grant.</p>\n        </li>\n        </ul>\n        <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n          <dd>\n              <p>Use this value to activate a grant
without replacing any member accounts\n              active grants for the same product.</p>\n          </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n          <dd>\n              <p>Use this value to activate a
```

grant and disable other active grants in any\n   member accounts
 for the same product. This action will also replace their\n   previously activated grants with this activated
grant.</p>\n   </dd>\n   </dl>"
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n   override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "com.amazonaws.licensemanager#OrganizationConfiguration": {
            "type": "structure",
            "members": {
              "EnableIntegration": {
                "target": "com.amazonaws.licensemanager#Boolean",
                "traits": {
                  "smithy.api#default": false,
                  "smithy.api#documentation":
 "<p>Enables Organizations integration.</p>",
                  "smithy.api#required": {}
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
            }
        },
        "com.amazonaws.licensemanager#PrincipalArnList": {
            "type": "list",
            "member": {
              "target": "com.amazonaws.licensemanager#Arn"
            },
            "traits": {
              "smithy.api#length": {
                "min": 1,
                "max": 1
              }
            }
        },
        "com.amazonaws.licensemanager#ProductInformation": {
            "type": "structure",
            "members": {
              "ResourceType": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {

                    "smithy.api#documentation": "<p>Resource type. The
 possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductInformationFilterList": {
                "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
                "traits": {
                    "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n   <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n   <p>The following filters and are supported when
the resource type \n   is <code>SSM_MANAGED</code>:</p>\n
      \n   \n   <p>\n   <code>Application Name</code> - The name of the
application.\n   Logical operator is <code>EQUALS</code>.</p>\n   \n   \n
<p>\n   <code>Application Publisher</code> - The publisher of the application.\n   Logical
operator is <code>EQUALS</code>.</p>\n   \n   \n   <p>\n
<code>Application Version</code> - The version of the application.\n   Logical operator is
<code>EQUALS</code>.</p>\n   \n   \n   <p>\n   <code>Platform
Name</code> - The name of the platform.\n   Logical operator is <code>EQUALS</code>.</p>\n
\n   \n   <p>\n   <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
      \n   \n   <p>\n   <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>. The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n   </p>\n   \n   \n   <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n   Logical operator is <code>NOT_EQUALS</code>.</p>\n   \n
\n   <p>\n   <code>License Included</code> - The type of license included.\n   Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n   Possible values are:
 <code>sql-server-enterprise</code> | \n   <code>sql-server-standard</code> | \n   <code>sql-server-
web</code> | \n   <code>windows-server-datacenter</code>.</p>\n   \n   \n
<p>The following filters and logical operators are supported when the resource type\n   is
<code>RDS</code>:</p>\n   \n   \n   <p>\n   <code>Engine Edition</code> -
The edition of the database engine.\n   Logical operator is <code>EQUALS</code>.\n   Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n   \n   \n   <p>\n   <code>License Pack</code> - The license
pack.\n   Logical operator is <code>EQUALS</code>.\n   Possible values are: <code>data
guard</code> | \n   <code>diagnostic pack sqlt</code> | \n
      <code>tuning pack sqlt</code> | \n   <code>ols</code> | \n   <code>olap</code>.</p>\n
   \n   ",
                    "smithy.api#required": {}
                }
            }
        },

```
      "traits": {
        "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilter": {
      "type": "structure",
      "members": {
        "ProductInformationFilterName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Filter name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductInformationFilterValue": {
          "target": "com.amazonaws.licensemanager#StringList",

      "traits": {
            "smithy.api#documentation": "<p>Filter value.</p>"
          }
        },
        "ProductInformationFilterComparator": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Logical operator.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes product information filters.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformationFilter"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformation"
      }
    },
    "com.amazonaws.licensemanager#ProvisionalConfiguration": {
      "type": "structure",
      "members": {
```

```
        "MaxTimeToLiveInMinutes": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#RateLimitExceededException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "RateLimitExceeded",
          "httpResponseCode": 429
        },
        "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 429
      }
    },
    "com.amazonaws.licensemanager#ReceivedMetadata": {
      "type": "structure",
      "members": {
        "ReceivedStatus": {
          "target": "com.amazonaws.licensemanager#ReceivedStatus",
          "traits": {
            "smithy.api#documentation": "<p>Received status.</p>"
          }
        },
        "ReceivedStatusReason": {
          "target": "com.amazonaws.licensemanager#StatusReasonMessage",
          "traits": {
            "smithy.api#documentation": "<p>Received
status reason.</p>"
          }
        },
        "AllowedOperations": {
          "target": "com.amazonaws.licensemanager#AllowedOperationList",
```

```
        "traits": {
          "smithy.api#documentation": "<p>Allowed operations.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
    }
  },
  "com.amazonaws.licensemanager#ReceivedStatus": {
    "type": "enum",
    "members": {
      "PENDING_WORKFLOW": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_WORKFLOW"
        }
      },
      "PENDING_ACCEPT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue":
"PENDING_ACCEPT"
        }
      },
      "REJECTED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "REJECTED"
        }
      },
      "ACTIVE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "ACTIVE"
        }
      },
      "FAILED_WORKFLOW": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "FAILED_WORKFLOW"
        }
      },
      "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DELETED"
        }
```

```
          },
          "DISABLED": {
            "target": "smithy.api#Unit",
             "traits": {
               "smithy.api#enumValue": "DISABLED"
             }
          },
          "WORKFLOW_COMPLETED": {
            "target": "smithy.api#Unit",
            "traits": {
               "smithy.api#enumValue": "WORKFLOW_COMPLETED"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanager#RedirectException": {
      "type": "structure",
      "members": {
        "Location": {
          "target": "com.amazonaws.licensemanager#Location",
          "traits": {
             "smithy.api#httpHeader": "Location"
          }
        },
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
         "smithy.api#error": "client",
        "smithy.api#httpError": 308
      }
    },
    "com.amazonaws.licensemanager#RejectGrant": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#RejectGrantRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#RejectGrantResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
          }
        },
        "com.amazonaws.licensemanager#RejectGrantRequest": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}

      }
        },
        "com.amazonaws.licensemanager#RejectGrantResponse": {
          "type": "structure",
          "members": {
            "GrantArn": {
              "target": "com.amazonaws.licensemanager#Arn",
              "traits": {
                "smithy.api#documentation": "<p>Grant ARN.</p>"
              }
            },
            "Status": {
```

```
                    "target": "com.amazonaws.licensemanager#GrantStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant status.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Grant version.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#RenewType":
{
            "type": "enum",
            "members": {
                "NONE": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "None"
                    }
                },
                "WEEKLY": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Weekly"
                    }
                },
                "MONTHLY": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "Monthly"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#ReportContext": {
            "type": "structure",
            "members": {
                "licenseConfigurationArns": {
                    "target": "com.amazonaws.licensemanager#ArnList",
                    "traits": {
```

```
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
    }
},
"com.amazonaws.licensemanager#ReportFrequency": {
    "type": "structure",
    "members": {
        "value": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
                "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n      The only supported value is <code>1</code>.</p>"
            }
        },
        "period": {
            "target": "com.amazonaws.licensemanager#ReportFrequencyType",

            "traits": {
                "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
    }
},
"com.amazonaws.licensemanager#ReportFrequencyType": {
    "type": "enum",
    "members": {
        "DAY": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "DAY"
            }
        },
        "WEEK": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "WEEK"
            }
        },
```

```
          "MONTH": {
            "target": "smithy.api#Unit",
            "traits":
{
              "smithy.api#enumValue": "MONTH"
            }
          }
        }
    },
    "com.amazonaws.licensemanager#ReportGenerator": {
      "type": "structure",
      "members": {
        "ReportGeneratorName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>"
          }
        },
        "ReportType": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
          }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
          }
        },
        "ReportFrequency": {
          "target": "com.amazonaws.licensemanager#ReportFrequency",
          "traits": {
            "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
          }
        },
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
          }
        },
        "LastRunStatus": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
          }
```

```
        },
        "LastRunFailureReason":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
            }
        },
        "LastReportGenerationTime": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
          }
        },
        "ReportCreatorAccount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",

      "traits": {
            "smithy.api#documentation": "<p>Description of the report generator.</p>"
          }
        },
        "S3Location": {
          "target": "com.amazonaws.licensemanager#S3Location",
          "traits": {
            "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
          }
        },
        "CreateTime": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Time the report was created.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
          }
```

```
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportGenerator"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorName": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "com.amazonaws.licensemanager#ReportType": {
      "type": "enum",
      "members": {
        "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
          }

        },
        "LICENSE_CONFIGURATION_USAGE_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationUsageReport"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportTypeList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportType"
      }
    },
    "com.amazonaws.licensemanager#ResourceInventory": {
      "type": "structure",
      "members": {
```

```
        "ResourceId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the resource.</p>"
          }
        },
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits":
{
            "smithy.api#documentation": "<p>Type of resource.</p>"
          }
        },
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },
        "Platform": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Platform of the resource.</p>"
          }
        },
        "PlatformVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
          }
        },
        "ResourceOwningAccountId":
{
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ResourceInventoryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ResourceInventory"
      }
```

```
        },
        "com.amazonaws.licensemanager#ResourceLimitExceededException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "ResourceLimitExceeded",

              "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#ResourceNotFoundException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "InvalidResource.NotFound",
              "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>The resource cannot be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#ResourceType": {
          "type": "enum",
          "members": {
            "EC2_INSTANCE": {
              "target": "smithy.api#Unit",
              "traits": {
                "smithy.api#enumValue": "EC2_INSTANCE"
              }
            },
            "EC2_HOST": {
              "target": "smithy.api#Unit",
              "traits": {
```

```
              "smithy.api#enumValue": "EC2_HOST"
            }
          },
          "EC2_AMI": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "EC2_AMI"
            }
          },
          "RDS": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "RDS"
            }
          },
          "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#S3Location": {
        "type": "structure",
        "members": {
          "bucket": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
            }
          },
          "keyPrefix": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
        }
      },
      "com.amazonaws.licensemanager#ServerInternalException": {
        "type": "structure",
        "members": {
          "Message": {
```

```
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "InternalError",
              "httpResponseCode": 500
            },
            "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
          }
        },
        "com.amazonaws.licensemanager#SignedToken": {
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 4096
            }
          }
        },
        "com.amazonaws.licensemanager#StatusReasonMessage": {
          "type": "string",
          "traits":
{
            "smithy.api#length": {
              "min": 0,
              "max": 400
            },
            "smithy.api#pattern": "^[\\s\\S]+$"
          }
        },
        "com.amazonaws.licensemanager#String": {
          "type": "string"
        },
        "com.amazonaws.licensemanager#StringList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#String"
          }
        },
        "com.amazonaws.licensemanager#Tag": {
          "type": "structure",
          "members": {
            "Key": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Tag key.</p>"
```

```
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation":
"<p>Tag value.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagKeyList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
      "com.amazonaws.licensemanager#TagList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Tag"
        }
      },
      "com.amazonaws.licensemanager#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#TagResourceResponse"
        },
        "errors": [
          {
            "target":
"com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
```

```
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits":
{
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>One or more tags.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#TokenData": {
      "type": "structure",
      "members": {
        "TokenId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token ID.</p>"
          }
        },
```

```
      "TokenType": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
        }
      },
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
        }
      },
      "ExpirationTime": {
        "target": "com.amazonaws.licensemanager#ISO8601DateTime",
        "traits": {
          "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
 format.</p>"
        }
      },
      "TokenProperties": {
        "target": "com.amazonaws.licensemanager#MaxSize3StringList",
        "traits": {
          "smithy.api#documentation": "<p>Data specified by the caller.</p>"
        }
      },
      "RoleArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
        }
      }
    },
    "traits": {

      "smithy.api#documentation": "<p>Describes a token.</p>"
    }
  },
  "com.amazonaws.licensemanager#TokenList": {
    "type": "list",
```

```
      "member": {
        "target": "com.amazonaws.licensemanager#TokenData"
      }
    },
    "com.amazonaws.licensemanager#TokenString": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 4096
        },
        "smithy.api#pattern": "^\\S+$"
      }
    },
    "com.amazonaws.licensemanager#TokenType": {
      "type": "enum",
      "members": {
        "REFRESH_TOKEN": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REFRESH_TOKEN"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
{
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    },
    "com.amazonaws.licensemanager#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UntagResourceResponse"
      },
```

```
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
       },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
          "smithy.api#required": {}
        }
      },
      "TagKeys": {
        "target": "com.amazonaws.licensemanager#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#UntagResourceResponse": {
    "type": "structure",
    "members": {},
```

```
        "traits": {
          "smithy.api#output": {}
        }
      },
    "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
      },
      "output": {

"target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
```

```
"LicenseConfigurationStatus": {
    "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
    "traits": {
        "smithy.api#documentation": "<p>New status of the license configuration.</p>"
    }
},
"LicenseRules": {
    "target": "com.amazonaws.licensemanager#StringList",
    "traits": {

        "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n configuration is licenseAffinityToHost.</p>"
    }
},
"LicenseCount": {
    "target": "com.amazonaws.licensemanager#BoxLong",
    "traits": {
        "smithy.api#documentation": "<p>New number of licenses managed by the license configuration.</p>"
    }
},
"LicenseCountHardLimit": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
    "traits": {
        "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
    }
},
"Name": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>New name of the license configuration.</p>"
    }
},
"Description": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>New description of the license configuration.</p>"
    }
},
"ProductInformationList": {
    "target": "com.amazonaws.licensemanager#ProductInformationList",
    "traits": {
        "smithy.api#documentation": "<p>New product information.</p>"
    }
},
"DisassociateWhenNotFound": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            }
        },
        "traits":
{
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
        "type": "structure",
        "members": {
            "LicenseManagerReportGeneratorArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
                    "smithy.api#required": {}
                }
            },
            "ReportGeneratorName": {
                "target": "com.amazonaws.licensemanager#ReportGeneratorName",
                "traits": {
                    "smithy.api#documentation": "<p>Name of the report generator.</p>",
                    "smithy.api#required": {}
                }
            },
            "Type": {
                "target": "com.amazonaws.licensemanager#ReportTypeList",
                "traits": {
                    "smithy.api#documentation":
"<p>Type of reports to generate. The following report types are supported:</p>\n        <ul>\n        <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n        </li>\n        <li>\n        <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n        </li>\n        </ul>",
                    "smithy.api#required": {}
                }
            },
            "ReportContext": {
                "target": "com.amazonaws.licensemanager#ReportContext",
                "traits": {
                    "smithy.api#documentation": "<p>The report context.</p>",
                    "smithy.api#required": {}
                }
            },
            "ReportFrequency": {
                "target": "com.amazonaws.licensemanager#ReportFrequency",
```

```
            "traits":
{
                "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                "smithy.api#required": {}
              }
            },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientRequestToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                "smithy.api#required": {}
              }
            },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
          }
        },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
```

```
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },


        {
          "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
        },
        {
          "target": "com.amazonaws.licensemanager#LicenseUsageException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n        <p>You can update the license specifications of AMIs, instances, and
hosts.\n        You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
      }
    },

    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
            "smithy.api#required": {}
          }
        },
        "AddLicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
          }
        },
        "RemoveLicenseSpecifications": {
          "target": "com.amazonaws.licensemanager#LicenseSpecifications",
          "traits": {
            "smithy.api#documentation":
"<p>ARNs of the license configurations to remove.</p>"
          }
```

```
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

```
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
                }
            },
            "SnsTopicArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
                }
            },
            "OrganizationConfiguration": {
                "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
                }
            },
            "EnableCrossAccountsDiscovery": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

        }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UsageOperation": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            }
        }
    },
    "com.amazonaws.licensemanager#ValidationException": {
```

```
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

    "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {
```

```go
	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// Current version
	of the license.
	//
	// This member is required.
	SourceVersion *string

	noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

	// Date when the license is deleted.
	DeletionDate *string

	// License status.
	Status types.LicenseDeletionStatus

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack,
	options); err != nil {
		return err
```

```go
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
    return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if
    err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
    middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
```

```go
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:        region,
ServiceID:     ServiceID,
OperationName: "DeleteLicense",
}
}
```

1.36 aws-aws-sdk-go-v2 1.14.8

1.36.1 Available under license :

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. Upgrading to Grafana
// Enterprise incurs additional fees. For more information, see Upgrade a
// workspace to Grafana Enterprise (https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-
Enterprise.html)
// .
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
```

```go
	if params == nil {
		params = &AssociateLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "AssociateLicense", params, optFns,
c.addOperationAssociateLicenseMiddlewares)
	if err != nil {
		return nil,
	err
	}

	out := result.(*AssociateLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type AssociateLicenseInput struct {

	// The type of license to associate with the workspace.
	//
	// This member is required.
	LicenseType types.LicenseType

	// The ID of the workspace to associate the license with.
	//
	// This member is required.
	WorkspaceId *string

	noSmithyDocumentSerde
}

type AssociateLicenseOutput struct {

	// A structure containing data about the workspace.
	//
	// This member is required.
	Workspace *types.WorkspaceDescription

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
```

```go
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err !=
nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack);
err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```go
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack,
 options); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

name: License Scan

on: [pull_request]

jobs:
 licensescan:

```
  name: License Scan
  runs-on: ubuntu-latest
  strategy:
   matrix:
     python-version: [3.9]


  steps:
    - name: Checkout target
      uses: actions/checkout@v2
      with:
       path: sdkbase
       ref: ${{ github.base_ref }}
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
       path: new-ref
       fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
       repository: nexB/scancode-toolkit
       path: scancode-toolkit
       fetch-depth: 1
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
       python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```go
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary
 *string

 noSmithyDocumentSerde
}

type CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
```

```go
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack);
	err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
	if err = addRetryMiddlewares(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
		return err
	}
	if err = addClientUserAgent(stack, options); err != nil {
		return err
```

```go
	}
	if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
		return err
	}
	if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
		return err
	}
	if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
		return
	err
	}
	if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
		return err
	}
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
	return &awsmiddleware.RegisterServiceMetadata{
		Region:       region,
		ServiceID:     ServiceID,
		OperationName: "CheckInLicense",
	}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
```

```
        "type": "integer"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n     Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {

          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
```

```
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest":
{
        "type": "structure",
        "members": {}
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>Lists the settings defined for Linux subscriptions discovery. The
settings include if\n      Organizations integration has been enabled, and
 which Regions data will be aggregated from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
```

```
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays
the aggregated data for Linux\n     subscriptions.</p>"
              }
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }

          },
            "AccountID": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
```

```
                    "smithy.api#documentation": "<p>The status of the instance.</p>"
                  }
                },
                "Region": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
                  }
                },
                "UsageOperation": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
 more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
                  }
                },
                "ProductCode": {
                  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
                  "traits": {
                    "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
                  }
                },
                "LastUpdatedTime": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The time in which the last discovery
 updated the instance details.</p>"
                  }
                },
                "SubscriptionName": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription being used by the instance.</p>"
                  }
                }
              },
              "traits": {
                "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
              }
            },
            "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
              "type": "list",
              "member": {
```

```
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
      ],

      "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
            "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
```

```
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                "x-api-key",
                "authorization",
                "x-amz-date",

  "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
          "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial
Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The
 AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
              "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
```

```
      "type": "Boolean"
    },
   "Endpoint": {
     "builtIn": "SDK::Endpoint",
     "required": false,
     "documentation": "Override the endpoint used to send this request",
     "type": "String"
   }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ],
      "type": "tree",
      "rules": [
        {

        "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseFIPS"
                },
                true
              ]
            }
          ],
          "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
          "type": "error"
        },
        {
          "conditions": [],
          "type": "tree",
          "rules": [
            {

            "conditions": [
                    {
                        "fn": "booleanEquals",
```

```
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],

    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
            ]
        }
    ]
},
{
    "conditions": [],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",

    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
```

```
                              "fn": "aws.partition",
                              "argv": [
                                {
                                  "ref": "Region"
                                }
                              ],

"assign": "PartitionResult"
                            }
                          ],
                          "type": "tree",
                          "rules": [
                            {
                              "conditions": [
                                {
                                  "fn": "booleanEquals",
                                  "argv": [
                                    {
                                      "ref": "UseFIPS"
                                    },
                                    true
                                  ]
                                },

                                {

                                  "fn": "booleanEquals",
                                  "argv": [
                                    {
                                      "ref": "UseDualStack"
                                    },
                                    true
                                  ]
                                }
                              ],
                              "type": "tree",
                              "rules": [
                                {
                                  "conditions": [
                                    {

                                  "fn": "booleanEquals",
                                      "argv": [
                                        true,
                                        {
                                          "fn": "getAttr",
                                          "argv": [
                                            {
                                              "ref": "PartitionResult"
```

```
                                            },
                                            "supportsFIPS"
                                        ]

        }
                                    ]
                                },
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },

                                        "supportsDualStack"
                                            ]
                                        }
                                    ]
                                }
                            ],
                            "type": "tree",
                            "rules": [
                                {
                                    "conditions": [],
                                    "type": "tree",
                                    "rules": [
                                        {

                    "conditions": [],
                                            "endpoint": {
                                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                        }
                                    ]
                                }
                            ]

                },
                            {
                                "conditions": [],
```

                                        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

                                        "type": "error"
                                    }
                                ]
                            },
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
                                            {

                "ref": "UseFIPS"
                                            },
                                            true
                                        ]
                                    }
                                ],
                                "type": "tree",
                                "rules": [
                                    {
                                        "conditions": [
                                            {
                                                "fn": "booleanEquals",
                                                "argv": [
                                                    true,

                {

                                                        "fn": "getAttr",
                                                        "argv": [
                                                            {
                                                                "ref": "PartitionResult"
                                                            },
                                                            "supportsFIPS"
                                                        ]
                                                    }
                                                ]
                                            }
                                        ],
                                        "type": "tree",

                "rules": [
                                            {
                                                "conditions": [],
                                                "type": "tree",
                                                "rules": [
                                                    {

```
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                        "properties": {},
                                        "headers": {}

                                    },
                                     "type": "endpoint"
                                }
                            ]
                        }
                    ]
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"
                }
            ]

    },
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "type": "tree",
                        "rules": [
                            {

                        "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
```

```
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [],
                        "type": "tree",
                        "rules": [
                            {
                                "conditions": [],

                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ]
                    }
                ]
            },

    {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ]
                },
                {
                    "conditions": [],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url":
"https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
```

```
                                    "type": "endpoint"
                                }
                            ]
                        }
                    ]
                }
            ]
        },
        {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
  }
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
         "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {

    "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
```

```
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint":
```

{

```
                    "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
```

```
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint":
{

                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
             "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
      },
      "params": {

       "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
```

```
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled
and DualStack enabled",
    "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
```

```
                            "UseFIPS": false,
                            "UseDualStack": false,
                            "Endpoint": "https://example.com"
                        }
                    },
                    {
                        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",

            "expect": {
                            "endpoint": {
                                "url": "https://example.com"
                            }
                        },
                        "params": {
                            "UseFIPS": false,
                            "UseDualStack": false,
                            "Endpoint": "https://example.com"
                        }
                    },
                    {
                        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                        "expect": {
                            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                        },
                        "params": {
                            "Region": "us-east-1",
                            "UseFIPS": true,
                            "UseDualStack": false,

        "Endpoint": "https://example.com"
                        }
                    },
                    {
                        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                        "expect": {
                            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                        },
                        "params": {
                            "Region": "us-east-1",
                            "UseFIPS": false,
                            "UseDualStack": true,
                            "Endpoint": "https://example.com"
                        }
                    },
                    {
                        "documentation": "Missing region",
                        "expect": {
```

```
                    "error": "Invalid Configuration: Missing Region"
                }

            }
        ],
        "version": "1.0"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type":
"operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with
commercial Linux\n      subscriptions.</p>",
          "smithy.api#http": {

      "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by
```

the name of\n   <code>AmiID</code> with an optional operator to see subscriptions that match, partially\n match, or don't match a certain Amazon Machine Image (AMI) ID.</p>\n   <p>The valid names for this filter are:</p>\n   \n   \n   <p>\n   <code>AmiID</code>\n   </p>\n   \n   \n   <p>\n   <code>InstanceID</code>\n   </p>\n   \n   \n   <p>\n   <code>AccountID</code>\n   </p>\n   \n   \n   <p>\n   <code>Status</code>\n   </p>\n   \n   \n   <p>\n <code>Region</code>\n   </p>\n   \n   \n   <p>\n <code>UsageOperation</code>\n   </p>\n   \n   \n   <p>\n   <code>ProductCode</code>\n   </p>\n   \n   \n   <p>\n   <code>InstanceType</code>\n   </p>\n   \n   \n   <p>The valid Operators for this filter are:</p>\n   \n   \n   <p>\n   <code>contains</code>\n   </p>\n   \n   \n   <p>\n   <code>equals</code>\n   </p>\n   \n   \n   <p>\n   <code>Notequal</code>\n   </p>\n   \n   "
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"

            }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation":
  "<p>An array that contains instance objects.</p>"
            }
          },

```
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have
linked your\n    organization, the returned results will include data aggregated across your accounts in\n
Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }

  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
```

```
"members": {
  "Filters": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n      </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n
    <ul>\n      <li>\n        <p>\n            <code>contains</code>\n        </p>\n      </li>\n
<li>\n        <p>\n            <code>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n
    <code>Notequal</code>\n        </p>\n      </li>\n    </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Token for the next set of results.</p>",
      "smithy.api#length": {

        "min": 1,
          "max": 16384
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
```

```
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
                    }
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "EQUAL",
                    "value": "Equal",
                    "documentation": "Equal operator"
                },
                {
                    "name": "NOT_EQUAL",
                    "value": "NotEqual",
                    "documentation": "Not equal operator"
                },
                {
                    "name": "CONTAINS",
                    "value": "Contains",
                    "documentation": "Contains operator"
                }
            ],
            "smithy.api#length": {
                "min": 1,
                "max": 20
            }
        }
    },

    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
```

```
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [

        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits":
{
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
```

```
        }
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"

          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included
with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services
Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as
OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux
```

subscription.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
          },

          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/UpdateServiceSettings"
            },

```
                    "smithy.api#idempotent": {}
                }
            },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest":
{
            "type": "structure",
            "members": {
                "LinuxSubscriptionsDiscovery": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                    "traits": {
                        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                        "smithy.api#required": {}
                    }
                },
                "LinuxSubscriptionsDiscoverySettings": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                    "traits": {
                        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",

    "smithy.api#required": {}
                    }
                },
                "AllowUpdate": {
                    "target": "smithy.api#Boolean",
                    "traits": {
                        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                    }
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
            "type": "structure",
            "members": {
                "LinuxSubscriptionsDiscovery": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                    "traits": {
                        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                    }
                },

         "LinuxSubscriptionsDiscoverySettings": {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                    "traits": {
                        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
```

settings include if Organizations\n   integration has been enabled, and which Regions data will be aggregated from.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
           "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
          "traits": {
           "smithy.api#documentation":
"<p>A message which details the Linux subscriptions service settings current status.</p>"
          }
        },
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
           "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n   subscriptions.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
      }
    }
  }
}

```go
// Code generated by smithy-go-codegen DO NOT EDIT.


package licensemanager

import (
 "context"
 "fmt"
```

```go
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
	if params == nil {
		params = &ListLicensesInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
	if err != nil {
		return nil, err
	}

	out := result.(*ListLicensesOutput)
	out.ResultMetadata = metadata
	return out, nil
}

type ListLicensesInput struct {

	// Filters to scope the results. The following filters are supported:
	//   - Beneficiary
	//   - ProductSKU
	//   - Fingerprint
	//
	//   - Status
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListLicensesOutput struct {
```

```go
	// License details.
	Licenses []types.License

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	if
	err := addProtocolFinalizerMiddlewares(stack, options, "ListLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
	if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
		return err
	}
	if err = addResolveEndpointMiddleware(stack, options); err != nil {
		return err
	}
	if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
		return err
	}
```

```
 if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
 err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:     region,
 ServiceID:
```

```go
        ServiceID,
    OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    //
    This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string
```

```go
    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
    options, "DisassociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addlegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
        return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
  return err
 }
 if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
 }
 if err = addRetryMiddlewares(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
```

```go
    }
    return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:        region,
        ServiceID:      ServiceID,
        OperationName: "DisassociateLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license. If the account that created the license is
// the same that is performing the check out, you must specify the account as the
// beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
    c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type
```

```
CheckoutLicenseInput struct {

    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string
```

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource
Name (ARN) of the checkout license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil
 {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
  return err
 }

```
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = addSetLegacyContextSigningOptionsMiddleware(stack);
err != nil {
 return err
 }
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
if err = addRequestResponseLogging(stack, options); err != nil {
```

```go
	return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
	return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName: "CheckoutLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
	if params == nil {
	params = &CheckoutBorrowLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
	if err != nil {
	return nil, err
	}

	out := result.(*CheckoutBorrowLicenseOutput)
	out.ResultMetadata = metadata
	return out, nil
}
```

```go
type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier
    that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see RFC 7518 Digital Signature with
    // RSASSA-PSS (https://tools.ietf.org/html/rfc7518#section-3.5) .
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string
```

```go
	// Date and time at which the license checkout is issued.
	IssuedAt *string

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License consumption token.
	LicenseConsumptionToken *string

	// Node ID.
	NodeId *string

	// Signed token.
	SignedToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
	if err != nil {
		return
	err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
		return err
	}
	if err = addSetLoggerMiddleware(stack, options); err != nil {
		return err
	}
	if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
		return err
	}
```

```
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
  return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
  return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
  return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
  return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
  return err
}
if err = addClientUserAgent(stack, options); err != nil {
  return
err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
}
if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
  return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
  return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
```

```go
    return err
   }
   if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
   }
   return nil
  }

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```go
type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide
    to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

    // Product name.
    //
    // This member is required.
    ProductName *string

    // Product SKU.
    //
```

```go
	// This member is required.
	ProductSKU *string

	// Date and time range during which the license is valid, in ISO8601-UTC format.
	//
	// This member is required.
	Validity
	*types.DatetimeRange

	// Information about the license.
	LicenseMetadata []types.Metadata

	noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

	// Amazon Resource Name (ARN) of the license.
	LicenseArn *string

	// License status.
	Status types.LicenseStatus

	// License version.
	Version *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
		return fmt.Errorf("add
	protocol finalizers: %v", err)
	}
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
 }
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
 return err
 }
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
 return err
 }
if err = addRetryMiddlewares(stack, options); err != nil {
 return err
 }
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
 return err
 }
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
 return err
 }
if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack);
err != nil {
 return err
 }
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
```

```go
	if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
		return err
	}
	if err = addRequestIDRetrieverMiddleware(stack); err != nil {
		return err
	}
	if err = addResponseErrorMiddleware(stack); err != nil {
		return err
	}
	if err = addRequestResponseLogging(stack, options); err != nil {
		return err
	}
	if err = addDisableHTTPSMiddleware(stack, options); err != nil {
		return err
	}
	return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
	return &awsmiddleware.RegisterServiceMetadata{
		Region:        region,
		ServiceID:     ServiceID,
		OperationName:
		"CreateLicense",
	}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
	"context"
	"fmt"
	awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
	"github.com/aws/aws-sdk-go-v2/aws/signer/v4"
	"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
	"github.com/aws/smithy-go/middleware"
	smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
	if params == nil {
		params = &GetLicenseInput{}
	}

	result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
```

```go
  c.addOperationGetLicenseMiddlewares)
  if err != nil {
   return nil, err
  }

  out := result.(*GetLicenseOutput)
  out.ResultMetadata = metadata
  return out, nil
 }

 type GetLicenseInput struct {

  // Amazon Resource Name (ARN) of the license.
  //
  // This member is required.
  LicenseArn *string

  // License version.
  Version
  *string

  noSmithyDocumentSerde
 }

 type GetLicenseOutput struct {

  // License details.
  License *types.License

  // Metadata pertaining to the operation's result.
  ResultMetadata middleware.Metadata

  noSmithyDocumentSerde
 }

 func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
  if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
   return err
  }
  err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
  if err != nil {
   return err
  }
  err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
  if err != nil {
   return err
  }
  if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
```

```go
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return
err
}
if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
return err
}
if err = addRetryMiddlewares(stack, options); err != nil {
return err
}
if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
return err
}
if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addOpGetLicenseValidationMiddleware(stack);
err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
```

```
err != nil {
 return err
 }
 if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
```

OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```go
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //   - Status
 //   - Fingerprint
 //   - IssuerName
```

```go
	//   - Beneficiary
	Filters []types.Filter

	// Amazon Resource Names (ARNs) of the licenses.
	LicenseArns []string

	// Maximum number of results to return in a single call.
	MaxResults *int32

	// Token for the next set of results.
	NextToken *string

	noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

	// Received license details.
	Licenses []types.GrantedLicense

	// Token for the next set of results.
	NextToken *string

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
    return err
  }
  if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
  }
  if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
    return err
  }
  if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
    return err
  }
  if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
  }
  if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
  }
  if err = addRetryMiddlewares(stack, options); err != nil {
    return err
  }
  if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
  }
  if err = awsmiddleware.AddRecordResponseTiming(stack);
  err != nil {
    return err
  }
  if err = addClientUserAgent(stack, options); err != nil {
    return err
  }
  if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
  }
  if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
  }
  if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
  }
  if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
  }
  if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
  }
  if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
```

```
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
      return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
      return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
      return err
    }
    return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region
  string) *awsmiddleware.RegisterServiceMetadata {
  return &awsmiddleware.RegisterServiceMetadata{
    Region:       region,
    ServiceID:    ServiceID,
    OperationName: "ListReceivedLicenses",
  }
}
```

<div align="center">

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

</div>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership
of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this
License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

   4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding
those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
```

```
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>Details about an Active Directory identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
```

```
                    {
                        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
                    }
                ],
                "traits": {
                    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p>\n        <note>\n            <p>Your estimated bill for charges on the number of users and related
costs will take 48\n        hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing
status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n
    monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n        </note>",
                "smithy.api#http": {
                    "code": 200,
                    "method": "POST",
                    "uri": "/user/AssociateUser"
                },
                "smithy.api#idempotent": {}
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
            "type": "structure",
            "members": {
                "Username": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                        "smithy.api#required": {}
                    }
                },
                "InstanceId": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The ID of the
EC2 instance, which provides user-based subscriptions.</p>",
                        "smithy.api#required": {}
                    }
                },
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#documentation": "<p>The identity provider of the user.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Domain": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The domain name of the user.</p>"
```

```
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the\n      resource.</p>",
            "smithy.api#error": "server"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider":
{
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deregisters the identity provider from providing user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
          "type": "structure",
```

"members": {
    "IdentityProviderSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
            "smithy.api#documentation":
"<p>Metadata that describes the results of an identity provider operation.</p>",
            "smithy.api#required": {}
        }
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```
          "uri": "/user/DisassociateUser"
        },
        "smithy.api#idempotent": {}
      }

    },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation":
 "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
```

```
                    }
                }
            }
        },

    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members": {
            "Attribute": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
                }
            },
            "Operation": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
                }
            },
            "Value": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Value of the filter.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used
 to return more specific results from a describe\n     operation. Filters can be used to match a set of resources by
specific criteria, such as tags,\n     attributes, or IDs.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
            "ActiveDirectoryIdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that details an Active Directory identity provider.</p>"
                }
            }
```

```
            },
            "traits": {
                "smithy.api#documentation":
    "<p>Details about an identity provider.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
            "type": "structure",
            "members": {
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Settings": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                    "traits": {
                        "smithy.api#documentation": "<p>An object that details the registered identity providers product
    related configuration\n     settings such as the subnets to provision VPC endpoints.</p>",
                        "smithy.api#required":
    {}
                    }
                },
                "Product": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Status": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The status of an identity provider.</p>",
                        "smithy.api#required": {}
                    }
                },
                "FailureMessage": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#documentation":
```

"<p>Describes an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary": {
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status
of an EC2 instance resource.</p>",
                "smithy.api#required": {}
              }
            },
            "Products": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",
                "smithy.api#required": {}
              }
            },
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
              }

```
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
            "type": "structure",
            "members": {
                "Username": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                        "smithy.api#required": {}
                    }
                },
                "InstanceId": {
                    "target": "smithy.api#String",
                    "traits":
{
                        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                        "smithy.api#required": {}
                    }
                },
                "IdentityProvider": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                    "traits": {
                        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Status": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                        "smithy.api#required": {}
                    }
                },
                "StatusMessage": {
                    "target": "smithy.api#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the user.</p>"
          }
        },
        "AssociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
          }
        },
        "DisassociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
```

```
    "type": "service",

  "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
      },

      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
```

```
            },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                "x-api-key",
                "authorization",
                "x-amz-date",
                "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"

    ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with\n      a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
                "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
                    "documentation": "The AWS region used to dispatch the request.",
                    "type": "String"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack
endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                    "type": "Boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
```

```
                        "required": true,
                        "default": false,
                        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                        "type": "Boolean"
                    },
                    "Endpoint": {
                        "builtIn": "SDK::Endpoint",
                        "required": false,
                        "documentation": "Override the endpoint used to send this request",
                        "type": "String"
                    }

        },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        },

                                        true
                                    ]
                                }
                            ],
                            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                            "type": "error"
                        },
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
```

```
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
                      },
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                }
              ]
            },
            {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [

{
                        "ref": "Region"
                      }
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "aws.partition",
                        "argv": [
                          {
                            "ref": "Region"
                          }
                        ],
```

```
                            "assign": "PartitionResult"
                          }
                        ],
                        "type": "tree",

              "rules": [
                          {
                            "conditions": [
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  {
                                    "ref": "UseFIPS"
                                  },
                                  true
                                ]
                              },
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  {
                                    "ref": "UseDualStack"

                                  },
                                  true
                                ]
                              }
                            ],
                            "type": "tree",
                            "rules": [
                              {
                                "conditions": [
                                  {
                                    "fn": "booleanEquals",
                                    "argv": [
                                      true,
                                      {
                                        "fn": "getAttr",

  "argv": [

                                          {
                                            "ref": "PartitionResult"
                                          },
                                          "supportsFIPS"
                                        ]
                                      }
                                    ]
                                  },
```

```
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
            "fn": "getAttr",
                                    "argv": [
                                      {
                                        "ref": "PartitionResult"
                                      },
                                      "supportsDualStack"
                                    ]
                                  }
                              ]
                            }
                          ],
                          "type": "tree",
                          "rules": [
                            {
            "conditions": [],
                                "endpoint": {
                                  "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
                                "type": "endpoint"
                              }
                          ]
                        },
                        {
                          "conditions": [],
                          "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                          "type": "error"
                        }
                      ]
                    },
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseFIPS"
                            },
```

```
                    true
                ]
            }
        ],
        "type": "tree",

  "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"

                                ]
                            }
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}

        },
                        "type": "endpoint"
                    }
                ]
            },
            {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"
            }
        ]
    },
```

```
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        },
```

```
                                {
                                    "conditions": [],
                                    "error": "DualStack is enabled but this partition does
not support DualStack",
                                    "type": "error"
                                }
                            ]
                        },
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ]
                }
            ]
        },
        {
            "conditions":
[],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation":
```

"For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },

          "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with
FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                  }
                },

```
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },

  "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1
with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
```

```
              "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                }
              },

        "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
                }
              },
```

```
      "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect":
{
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack":
true
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
```

```
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "error": "FIPS and DualStack are enabled, but
this partition does not support one or both"
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }

      },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {

"documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
             "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
```

```
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the identity providers for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/ListIdentityProviders"
        },
        "smithy.api#paginated":
{

          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "IdentityProviderSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse":
{

      "type": "structure",
      "members": {
        "IdentityProviderSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
          "traits": {
```

```
              "smithy.api#documentation": "<p>Metadata that describes the list identity providers operation.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
```

```
                "uri": "/instance/ListInstances"
              },
              "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
          "type":
"structure",
          "members": {
            "InstanceSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list instances operation.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
```

```
                }
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",
              "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
              },
              "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
```

```
                }
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
              "type": "structure",
              "members": {
                "Product": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>The name
of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                  }
                },
                "IdentityProvider": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                  "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                  }
                },
                "MaxResults": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                  "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                  }
                },
                "Filters": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                  "traits": {

                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match
one or\n     more sets of key-value pairs that you specify.</p>"
                  }
                },
                "NextToken": {
                  "target": "smithy.api#String",
                  "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                  }
                }
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
              "type": "structure",
              "members": {
                "ProductUserSummaries": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },

    "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
```

```
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated":
{
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceUserSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n      more sets of key-value pairs that you specify.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Token for
the next set of results.</p>"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {

        "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
```

```
        },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of a product for a user.</p>",
                    "smithy.api#required": {}
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for a product for a user.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the user.</p>"
                }
            },
            "SubscriptionStartDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The
start date of a subscription.</p>"
                }
            },
            "SubscriptionEndDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The end date of a subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The summary of the user-based subscription products for a user.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity
provider.</p>",
                "smithy.api#required": {}
              }
```

```
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the\n      subnets to provision VPC endpoints.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
        "type": "string",
```

```
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
        "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation":
"<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                    "min": 1
                },
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in\n     your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the\n     subnets to provision VPC endpoints, and the security group ID that
is associated with the VPC\n     endpoints. The security group should permit inbound TCP port 1688
communication from resources\n     in the VPC.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p>\n    <note>\n      <p>Your estimated bill for charges on the number of users and related costs
will take 48\n    hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status)
in Amazon Web Services Billing.
  For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your\n      monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p>\n    </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StartProductSubscription"
          }
        }
      },
```

```
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
```

```
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"

        },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
```

```
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required":
{}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the user.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation":
"<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StringList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "subnet-[a-z0-9]{8,17}"
          }
```

```
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {


            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity\n      provider.</p>",
          "smithy.api#http": {
            "code": 200,
```

```
              "method": "POST",
              "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
               "smithy.api#required": {}
             }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n     update any combination of settings in a single operation such as the:</p>\n
<ul>\n       <li>\n            <p>Subnets which you want to add to provision VPC endpoints.</p>\n        </li>\n
      <li>\n            <p>Subnets which you want to remove the VPC endpoints from.</p>\n        </li>\n
<li>\n            <p>Security group ID which permits traffic to the VPC endpoints.</p>\n        </li>\n       </ul>",
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
```

```
      "members": {
          "AddSubnets": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that\n     require connectivity to activation servers.</p>",
              "smithy.api#required": {}
            }
          },
          "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID
 that allows inbound TCP port 1688 communication between resources in\n     your VPC and the VPC endpoints for
activation servers.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as\n     the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
}
{
  "smithy": "2.0",
  "metadata": {
```

```json
    "suppressions": [
      {
        "id": "HttpMethodSemantics",
        "namespace": "*"
      },
      {
        "id": "HttpResponseCodeSemantics",
        "namespace": "*"
      },
      {
        "id": "PaginatedTrait",
        "namespace": "*"
      },
      {
        "id": "HttpHeaderTrait",
        "namespace": "*"
      },
      {
        "id": "HttpUriConflict",
        "namespace": "*"
      },
      {
        "id": "Service",
        "namespace": "*"
      }
    ]
  },
  "shapes": {
    "com.amazonaws.licensemanager#AWSLicenseManager": {
      "type": "service",
      "version": "2018-08-01",
      "operations": [
        {
          "target": "com.amazonaws.licensemanager#AcceptGrant"
        },

        {
          "target": "com.amazonaws.licensemanager#CheckInLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CheckoutLicense"
        },
        {
          "target": "com.amazonaws.licensemanager#CreateGrant"
        },
```

```
      {
        "target": "com.amazonaws.licensemanager#CreateGrantVersion"
      },
      {
        "target": "com.amazonaws.licensemanager#CreateLicense"
      },
      {
        "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
      },
      {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
      },
      {
        "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
       },
      {
        "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
      },
      {
        "target": "com.amazonaws.licensemanager#CreateToken"
      },
      {
        "target": "com.amazonaws.licensemanager#DeleteGrant"
      },
      {
        "target": "com.amazonaws.licensemanager#DeleteLicense"
      },
      {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
      },
      {
        "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
      },
      {
        "target": "com.amazonaws.licensemanager#DeleteToken"
      },
      {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
      }

    },
      {
        "target": "com.amazonaws.licensemanager#GetAccessToken"
      },
      {
        "target": "com.amazonaws.licensemanager#GetGrant"
      },
      {
        "target": "com.amazonaws.licensemanager#GetLicense"
```

    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
    },
    {
      "target": "com.amazonaws.licensemanager#GetLicenseUsage"
    },
    {
      "target": "com.amazonaws.licensemanager#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
    },
    {
      "target": "com.amazonaws.licensemanager#ListDistributedGrants"
    },
    {
      "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenses"
    },
    {
      "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
    },
    {

      "target": "com.amazonaws.licensemanager#ListLicenseVersions"
    },
    {
      "target": "com.amazonaws.licensemanager#ListReceivedGrants"
    },
    {

```
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
      },
      {
        "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
      },
      {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
      },
      {
        "target": "com.amazonaws.licensemanager#ListResourceInventory"
      },
      {
        "target": "com.amazonaws.licensemanager#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licensemanager#ListTokens"
      },
      {
        "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
       },
      {
        "target": "com.amazonaws.licensemanager#RejectGrant"
      },
      {
        "target": "com.amazonaws.licensemanager#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UntagResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
      },
      {
        "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager",
        "arnNamespace": "license-manager",
        "cloudFormationName": "LicenseManager",
        "cloudTrailEventSource": "licensemanager.amazonaws.com",
```

```
            "endpointPrefix": "license-manager"
          },
          "aws.auth#sigv4": {
            "name": "license-manager"
          },
          "aws.protocols#awsJson1_1": {},
          "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n      Amazon Web Services accounts and on-premises servers.</p>",
          "smithy.api#title": "AWS License Manager",
          "smithy.api#xmlNamespace": {
            "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
          },
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "Boolean"
              },
              "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send
 this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
                "type": "Boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "String"
              }
            },
            "rules": [
              {
                "conditions": [
```

```
                    {
                      "fn": "isSet",
                      "argv": [
                        {
                          "ref": "Endpoint"
                        }
                      ]

                    }
                  ],
                  "type": "tree",
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseFIPS"
                            },
                            true
                          ]
                        }
                      ],
                      "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                      "type": "error"
                    },
                    {

    "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }
                      ],
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": {
                          "ref": "Endpoint"
```

```
                },
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ]
        },
        {
          "conditions": [
            {
              "fn": "isSet",
              "argv": [
                {
                  "ref": "Region"
                }
              ]
            }
          ],
          "type": "tree",
          "rules": [
            {
              "conditions": [
                {
                  "fn": "aws.partition",
                  "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref":
"UseFIPS"
                        },
                        true
                      ]
                    },
                    {
```

```
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "type": "tree",
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]

                      }
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },

"supportsDualStack"

                        ]
                      }
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
```

```
                              {
                                "conditions": [],
                                "endpoint": {
                                   "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                   "properties": {},
                                   "headers": {}

                              },
                                "type": "endpoint"
                              }
                           ]
                        },
                        {
                           "conditions": [],
                           "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                           "type": "error"
                        }
                     ]
                  },
                  {
                     "conditions": [
                        {
                           "fn": "booleanEquals",

                  "argv": [
                              {
                                 "ref": "UseFIPS"
                              },
                              true
                           ]
                        }
                     ],
                     "type": "tree",
                     "rules": [
                        {
                           "conditions": [
                              {
                                 "fn": "booleanEquals",
                                 "argv": [
                                    true,

               {

                                    "fn": "getAttr",
                                    "argv": [
                                       {
                                          "ref": "PartitionResult"
```

```json
                            },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      }
                    ],
                    "type": "tree",
                    "rules": [

                      {
                              "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                              },
                              "type": "endpoint"
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "error": "FIPS is enabled but this
partition does not support FIPS",
                    "type": "error"
                  }
                ]
              },
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "type": "tree",
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
```

                              true,
                              {
                                "fn": "getAttr",
                                "argv": [
                                  {
                                      "ref": "PartitionResult"
                                  },

    "supportsDualStack"

                                ]
                              }
                            ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
                          {
                              "conditions": [],
                              "endpoint": {
                                "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}

                    },
                              "type": "endpoint"
                          }
                        ]
                      },
                      {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                      }
                    ]
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ]
              }
            ]

```
              },
              {
                 "conditions": [],
                 "error": "Invalid Configuration: Missing Region",
                 "type": "error"
              }
           ]
        },
        "smithy.rules#endpointTests": {
           "testCases": [
              {
                 "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
                 "expect": {
                    "endpoint":

{

                       "url": "https://license-manager.af-south-1.amazonaws.com"
                    }
                 },
                 "params": {
                    "Region": "af-south-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                 }
              },
              {
                 "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
                 "expect": {
                    "endpoint": {
                       "url": "https://license-manager.ap-east-1.amazonaws.com"
                    }
                 },
                 "params": {
                    "Region": "ap-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                 }
              },
              {
                 "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
                 "expect": {
                    "endpoint": {
                       "url": "https://license-manager.ap-northeast-1.amazonaws.com"
                    }
                 },
                 "params": {
                    "Region": "ap-northeast-1",
                    "UseFIPS": false,
                    "UseDualStack": false
```

```
          }
        },
        {
          "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-northeast-3.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-northeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }

        },
        {
          "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-south-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-south-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-1.amazonaws.com"
```

```
    }
          },
          "params": {
            "Region": "ap-southeast-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-2.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-2",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

  "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ap-southeast-3.amazonaws.com"
            }
          },
          "params": {
            "Region": "ap-southeast-3",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager.ca-central-1.amazonaws.com"
            }
          },

       "params": {
            "Region": "ca-central-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
```

```
            },
            {
              "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.eu-central-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "eu-central-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region eu-north-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.eu-north-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "eu-north-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.eu-south-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "eu-south-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.eu-west-1.amazonaws.com"
                }
```

```
      },
      "params": {
        "Region": "eu-west-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
      "expect": {

        "endpoint": {
          "url": "https://license-manager.eu-west-2.amazonaws.com"
        }
      },
      "params": {
        "Region": "eu-west-2",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.eu-west-3.amazonaws.com"
        }
      },
      "params": {
        "Region": "eu-west-3",
        "UseFIPS": false,

"UseDualStack": false
      }
    },
    {
      "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.me-south-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "me-south-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
```

```
{
    "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.sa-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "sa-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }

},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-east-2.amazonaws.com"
        }
```

```
        },
        "params": {
          "Region": "us-east-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-east-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-2",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region
us-west-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-west-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-west-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-west-1.amazonaws.com"
          }
        },
        "params": {

        "Region": "us-west-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
```

```json
      {
        "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-west-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-west-2",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-west-2.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-west-2",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,

          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-east-1.api.aws"
          }
        },
```

```
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
                }
              },
              "params": {
                "Region": "cn-north-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
                }
              },
              "params": {
                "Region": "cn-northwest-1",
                "UseFIPS": false,
                "UseDualStack": false

      }
            },
            {
              "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
                }
              },
              "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
```

```
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },

      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
```

```
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager.us-gov-west-1.amazonaws.com"
                        }
                    },
                    "params": {
                        "Region": "us-gov-west-1",
                        "UseFIPS": false,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation":
"For region us-gov-west-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
                        }
                    },
                    "params": {
                        "Region": "us-gov-west-1",
                        "UseFIPS": true,
                        "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://license-manager-fips.us-gov-east-1.api.aws"
                        }
                    },

    "params": {
                        "Region": "us-gov-east-1",
                        "UseFIPS": true,
                        "UseDualStack": true
                    }
                },
                {
                    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
```

```
        "endpoint": {
          "url": "https://license-manager.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
```

```
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "error": "DualStack is enabled
but this partition does not support DualStack"
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},

{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
},

"params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
```

```
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "error": "DualStack is enabled but this partition does not support DualStack"
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

    "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,

    "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
```

              },
              "params": {
                 "Region": "us-east-1",
                 "UseFIPS": true,
                 "UseDualStack": false,
                 "Endpoint": "https://example.com"
              }
           },
           {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                 "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                 "Region": "us-east-1",
                 "UseFIPS": false,
                 "UseDualStack": true,
                 "Endpoint": "https://example.com"
              }
           },
           {
              "documentation": "Missing region",
              "expect": {
                 "error": "Invalid Configuration: Missing Region"
              }
           }
        ],
        "version": "1.0"
     }
  }
},
"com.amazonaws.licensemanager#AcceptGrant": {
  "type": "operation",
  "input": {
     "target": "com.amazonaws.licensemanager#AcceptGrantRequest"

},
  "output": {
     "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
  },
  "errors": [
     {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
     },
     {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
     },

```
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"

        }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
        }
    },
    "com.amazonaws.licensemanager#AcceptGrantRequest": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#AcceptGrantResponse": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Grant
ARN.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
```

```
        "traits": {
          "smithy.api#documentation": "<p>Grant status.</p>"
        }
      },
      "Version": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Grant version.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#AccessDeniedException": {
    "type": "structure",
    "members": {
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "aws.protocols#awsQueryError": {

"code": "ServiceAccessDenied",
        "httpResponseCode": 401
      },
      "smithy.api#documentation": "<p>Access to resource denied.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 401
    }
  },
  "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
    "type": "enum",
    "members": {
      "DISTRIBUTED_GRANTS_ONLY": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
        }
      },
      "ALL_GRANTS_PERMITTED_BY_ISSUER": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
        }
      }
```

```
        }
      },
      "com.amazonaws.licensemanager#AllowedOperation": {

      "type": "enum",
        "members": {
          "CREATE_GRANT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CreateGrant"
            }
          },
          "CHECKOUT_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CheckoutLicense"
            }
          },
          "CHECKOUT_BORROW_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CheckoutBorrowLicense"
            }
          },
          "CHECK_IN_LICENSE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CheckInLicense"
            }
          },
          "EXTEND_CONSUMPTION_LICENSE":
{
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ExtendConsumptionLicense"
            }
          },
          "LIST_PURCHASED_LICENSES": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ListPurchasedLicenses"
            }
          },
          "CREATE_TOKEN": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "CreateToken"
            }
```

```
          }
        }
      },
      "com.amazonaws.licensemanager#AllowedOperationList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#AllowedOperation"
        },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max":
7
          }
        }
      },
      "com.amazonaws.licensemanager#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 2048
          },
          "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
        }
      },
      "com.amazonaws.licensemanager#ArnList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Arn"
        }
      },
      "com.amazonaws.licensemanager#AuthorizationException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError":
{
            "code": "AuthorizationFailure",
            "httpResponseCode": 403
          },
          "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to perform the action. Check the IAM\n        policy associated with this account.</p>",
```

```
            "smithy.api#error": "client",
            "smithy.api#httpError": 403
          }
      },
    "com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
        "type": "structure",
        "members": {
          "LastRunTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes automated discovery.</p>"
        }
    },

  "com.amazonaws.licensemanager#Boolean": {
        "type": "boolean",
        "traits": {
          "smithy.api#default": false
        }
    },
    "com.amazonaws.licensemanager#BorrowConfiguration": {
        "type": "structure",
        "members": {
          "AllowEarlyCheckIn": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxTimeToLiveInMinutes": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
              "smithy.api#required": {}
            }
          }

},
        "traits": {
          "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
        }
    },
```

```
"com.amazonaws.licensemanager#BoxBoolean": {
    "type": "boolean"
},
"com.amazonaws.licensemanager#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licensemanager#BoxLong": {
    "type": "long"
},
"com.amazonaws.licensemanager#CheckInLicense": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
    }
```

```
    },
    "com.amazonaws.licensemanager#CheckInLicenseRequest":
{
        "type": "structure",
        "members": {
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License consumption token.</p>",
              "smithy.api#required": {}
            }
          },
          "Beneficiary": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License beneficiary.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#CheckInLicenseResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
```

```
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
                },
                {
                    "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RedirectException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
            }
        },
        "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
            "type": "structure",
            "members":
{
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Entitlements": {
                    "target": "com.amazonaws.licensemanager#EntitlementDataList",
                    "traits": {
                        "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
                        "smithy.api#required": {}
```

```
            }
          },
          "DigitalSignatureMethod": {
            "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
            "traits": {
              "smithy.api#documentation": "<p>Digital signature method. The possible
value is JSON Web Signature (JWS) algorithm PS384.\n        For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>",
              "smithy.api#required": {}
            }
          },
          "NodeId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Node ID.</p>"
            }
          },
          "CheckoutMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about constraints.</p>"
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "LicenseConsumptionToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
                    "smithy.api#documentation": "<p>License consumption token.</p>"
                }
            },
            "EntitlementsAllowed":
{
                "target": "com.amazonaws.licensemanager#EntitlementDataList",
                "traits": {
                    "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
                }
            },
            "NodeId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Node ID.</p>"
                }
            },
            "SignedToken": {
                "target": "com.amazonaws.licensemanager#SignedToken",
                "traits": {
                    "smithy.api#documentation": "<p>Signed token.</p>"
                }
            },
            "IssuedAt": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
                }
            },
            "Expiration": {
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
                }
            },
            "CheckoutMetadata": {
                "target": "com.amazonaws.licensemanager#MetadataList",
                "traits": {
                    "smithy.api#documentation": "<p>Information about constraints.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CheckoutLicense": {
        "type": "operation",
```

```
"input": {
    "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
},
"output": {
    "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
},
"errors": [
    {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
        "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
    },
    {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
        "target": "com.amazonaws.licensemanager#RedirectException"
    },
    {
        "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
    },
    {
        "target":
"com.amazonaws.licensemanager#ServerInternalException"
    },
    {
        "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
    },
    {
        "target": "com.amazonaws.licensemanager#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Checks out the specified license.</p>\n        <note>\n            <p>If the
account that created the license is the same that is performing the check out, you must\n            specify the account
as the beneficiary.</p>\n        </note>"
}
},
"com.amazonaws.licensemanager#CheckoutLicenseRequest": {
    "type": "structure",
    "members": {
```

```
      "ProductSKU": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product
SKU.</p>",
          "smithy.api#required": {}
        }
      },
      "CheckoutType": {
        "target": "com.amazonaws.licensemanager#CheckoutType",
        "traits": {
          "smithy.api#documentation": "<p>Checkout type.</p>",
          "smithy.api#required": {}
        }
      },
      "KeyFingerprint": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
          "smithy.api#required": {}
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementDataList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>",

    "smithy.api#required": {}
        }
      },
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientToken",
        "traits": {
          "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
          "smithy.api#required": {}
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License beneficiary.</p>"
        }
      },
      "NodeId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Node ID.</p>"
```

```
                    }
                }
            },
            "traits": {

        "smithy.api#input": { }
            }
        },
        "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
            "type": "structure",
            "members": {
                "CheckoutType": {
                    "target": "com.amazonaws.licensemanager#CheckoutType",
                    "traits": {
                        "smithy.api#documentation": "<p>Checkout type.</p>"
                    }
                },
                "LicenseConsumptionToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License consumption token.</p>"
                    }
                },
                "EntitlementsAllowed": {
                    "target": "com.amazonaws.licensemanager#EntitlementDataList",
                    "traits": {
                        "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
                    }
                },
                "SignedToken": {
                    "target": "com.amazonaws.licensemanager#SignedToken",
                    "traits": {
                        "smithy.api#documentation": "<p>Signed token.</p>"
                    }
                },
                "NodeId": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Node ID.</p>"
                    }
                },
                "IssuedAt": {
                    "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
                    }
                },
                "Expiration": {
```

```
                    "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                    "traits": {
                        "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
                    }
                },
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#CheckoutType": {
            "type": "enum",
            "members": {
                "PROVISIONAL": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "PROVISIONAL"
                    }
                },
                "PERPETUAL": {
                    "target": "smithy.api#Unit",
                    "traits": {
                        "smithy.api#enumValue": "PERPETUAL"
                    }
                }
            }
        },
        "com.amazonaws.licensemanager#ClientRequestToken": {
            "type": "string",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 36
                }
            }
        },
        "com.amazonaws.licensemanager#ClientToken": {
            "type": "string",
            "traits": {
                "smithy.api#length": {
                    "min": 0,
```

```
          "max": 2048
        },
        "smithy.api#pattern": "^\\S+$"
      }
    },
    "com.amazonaws.licensemanager#ConflictException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "ConflictException",
          "httpResponseCode":
409
        },
        "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 409
      }
    },
    "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
      "type": "structure",
      "members": {
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
          }
        },
        "ConsumedLicenses": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about license consumption.</p>"
      }
    },
    "com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
      "type": "list",
      "member": {
```

```
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
        }
      },
      "com.amazonaws.licensemanager#ConsumptionConfiguration": {
        "type": "structure",
        "members": {
          "RenewType": {
            "target": "com.amazonaws.licensemanager#RenewType",
            "traits": {
              "smithy.api#documentation": "<p>Renewal frequency.</p>"
            }
          },
          "ProvisionalConfiguration": {
            "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
            }
          },
          "BorrowConfiguration": {
            "target": "com.amazonaws.licensemanager#BorrowConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateGrant": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateGrantRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateGrantResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target":
"com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
```

```
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n        entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
 information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateGrantRequest": {
      "type": "structure",
      "members": {
        "ClientToken": {
          "target": "com.amazonaws.licensemanager#ClientToken",
          "traits": {
            "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
            "smithy.api#required": {}
          }
        },
        "GrantName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Grant name.</p>",
            "smithy.api#required": {}

  }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
            "smithy.api#required": {}
          }
        },
        "Principals": {
```

```
                "target": "com.amazonaws.licensemanager#PrincipalArnList",
                "traits": {
                    "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an
Amazon Resource Name\n        (ARN):</p>\n    <ul>\n        <li>\n            <p>An Amazon Web Services
account, which includes only the account specified.</p>\n        </li>\n    </ul>\n    <ul>\n        <li>\n
    <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n        </li>\n    </ul>\n
<ul>\n
        <li>\n            <p>An organization, which will include all accounts across your organization.</p>\n
</li>\n    </ul>",
                    "smithy.api#required": {}
                }
            },
            "HomeRegion": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Home Region of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "AllowedOperations": {
                "target": "com.amazonaws.licensemanager#AllowedOperationList",
                "traits": {
                    "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

    "com.amazonaws.licensemanager#CreateGrantResponse": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>"
                }
            },
            "Version": {
```

```
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant version.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#CreateGrantVersion":
{
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>"
        }
    },
```

```
"com.amazonaws.licensemanager#CreateGrantVersionRequest": {
    "type": "structure",
    "members": {
        "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
                "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",

    "smithy.api#required": {}
            }
        },
        "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                "smithy.api#required": {}
            }
        },
        "GrantName": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Grant name.</p>"
            }
        },
        "AllowedOperations": {
            "target": "com.amazonaws.licensemanager#AllowedOperationList",
            "traits": {
                "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
            }
        },
        "StatusReason": {
            "target": "com.amazonaws.licensemanager#StatusReasonMessage",
            "traits": {
                "smithy.api#documentation": "<p>Grant status reason.</p>"
            }
        },
        "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Current version of the grant.</p>"
            }
```

```
          },
          "Options": {
            "target": "com.amazonaws.licensemanager#Options",
            "traits": {
              "smithy.api#documentation": "<p>The options specified for the grant.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateGrantVersionResponse": {
        "type": "structure",
        "members": {
          "GrantArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Grant ARN.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
              "smithy.api#documentation": "<p>Grant status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>New version of the grant.</p>"
            }
          }
        },
        "traits":
{
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
        },
        "errors": [
```

```
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#RedirectException"
       },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a license.</p>"
    }
  },
  "com.amazonaws.licensemanager#CreateLicenseConfiguration": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
       },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
```

```
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a license configuration.</p>\n        <p>A license configuration
is an abstraction of a customer license agreement that can be\n        consumed and enforced by License Manager.
Components include specifications for the license\n        type (licensing by instance, socket, CPU, or vCPU),
allowed tenancy (shared tenancy, \n        Dedicated Instance, Dedicated Host, or
 all of these), license affinity  to host (how long a \n        license must be associated with a host), and the number of
licenses purchased and used.</p>"
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the license configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "Description": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Description of the license configuration.</p>"
          }
        },
        "LicenseCountingType": {
          "target": "com.amazonaws.licensemanager#LicenseCountingType",
          "traits": {
            "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseCount": {
          "target": "com.amazonaws.licensemanager#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
          }
        },
        "LicenseCountHardLimit": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used.
```

Exceeding a hard limit\n   blocks the launch of new instances.</p>"
                }
            },
            "LicenseRules": {
                "target":
 "com.amazonaws.licensemanager#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example,
#allowedTenancy=EC2-DedicatedHost). The available rules \n   vary by dimension, as follows.</p>\n
\n   \n   <p>\n   <code>Cores</code> dimension: <code>allowedTenancy</code> |\n
        <code>licenseAffinityToHost</code> |\n   <code>maximumCores</code> |
<code>minimumCores</code>\n   </p>\n   \n   \n   <p>\n
<code>Instances</code> dimension: <code>allowedTenancy</code> | \n   <code>maximumCores</code> |
<code>minimumCores</code> |\n   <code>maximumSockets</code> | <code>minimumSockets</code> |\n
        <code>maximumVcpus</code> | <code>minimumVcpus</code>\n   </p>\n   \n
\n
 <p>\n   <code>Sockets</code> dimension: <code>allowedTenancy</code> | \n
<code>licenseAffinityToHost</code> |\n   <code>maximumSockets</code> |
<code>minimumSockets</code>\n   </p>\n   \n   \n   <p>\n
<code>vCPUs</code> dimension: <code>allowedTenancy</code> | \n
<code>honorVcpuOptimization</code> | \n   <code>maximumVcpus</code> |
<code>minimumVcpus</code>\n   </p>\n   \n   \n   <p>The unit for
<code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n   values for
<code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n
<code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n
<code>True</code> and <code>False</code>.</p>"
                }
            },

    "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                    "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
                }
            },
            "DisassociateWhenNotFound": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            },
            "ProductInformationList": {
                "target": "com.amazonaws.licensemanager#ProductInformationList",
                "traits": {
                    "smithy.api#documentation": "<p>Product information.</p>"
                }
            }

```
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
```

```
        "traits": {
          "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {

            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting
the license type for.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information that identifies the license type you are converting
from. \n\n      For the structure of the source license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
              "smithy.api#required": {}
            }
          },
          "DestinationLicenseContext":
 {
              "target": "com.amazonaws.licensemanager#LicenseConversionContext",
              "traits": {
                "smithy.api#documentation": "<p>Information that identifies the license type you are converting to.
For the structure of the destination license, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/conversion-procedures.html#conversion-cli\">Convert a license type using the CLI </a> in
the <i>License Manager User Guide</i>.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
        "type": "structure",
        "members": {
          "LicenseConversionTaskId": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
```

```
    "traits": {
            "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
        }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    },
    {
      "target": "com.amazonaws.licensemanager#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a report generator.</p>"
  }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
```

```
        "type": "structure",
        "members": {
          "ReportGeneratorName":
{
            "target": "com.amazonaws.licensemanager#ReportGeneratorName",
            "traits": {
              "smithy.api#documentation": "<p>Name of the report generator.</p>",
              "smithy.api#required": {}
            }
          },
          "Type": {
            "target": "com.amazonaws.licensemanager#ReportTypeList",
            "traits": {
              "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n        <ul>\n           <li>\n              <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n           </li>\n           <li>\n              <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n           </li>\n
</ul>",
              "smithy.api#required": {}

          }
          },
          "ReportContext": {
            "target": "com.amazonaws.licensemanager#ReportContext",
            "traits": {
              "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
              "smithy.api#required": {}
            }
          },
          "ReportFrequency": {
            "target": "com.amazonaws.licensemanager#ReportFrequency",
            "traits": {
              "smithy.api#documentation": "<p>Frequency by which reports are generated.  Reports can be
generated daily, monthly, or weekly.</p>",
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientRequestToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique,
case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
```

```
          "traits": {
            "smithy.api#documentation": "<p>Description of the report generator.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
      "type": "structure",
      "members": {
        "LicenseManagerReportGeneratorArn":
{
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#CreateLicenseRequest": {
      "type": "structure",
      "members": {
        "LicenseName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product
name.</p>",
            "smithy.api#required": {}
          }
```

```
      },
      "ProductSKU": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product SKU.</p>",
          "smithy.api#required": {}
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#Issuer",
        "traits": {
          "smithy.api#documentation": "<p>License issuer.</p>",
          "smithy.api#required": {}
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region for the license.</p>",
          "smithy.api#required": {}

  }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
          "smithy.api#required": {}
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>",
          "smithy.api#required": {}
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License beneficiary.</p>",
          "smithy.api#required": {}
        }
       },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
```

```
            "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseMetadata": {
            "target": "com.amazonaws.licensemanager#MetadataList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license.</p>"
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>License version.</p>"
            }
          }
        },
        "traits": {
```

```
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersion": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
        }
      },
      "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target":
"com.amazonaws.licensemanager#Arn",
            "traits": {
```

```
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License name.</p>",
          "smithy.api#required": {}
        }
      },
      "ProductName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product name.</p>",
          "smithy.api#required": {}
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#Issuer",
        "traits":

{

          "smithy.api#documentation": "<p>License issuer.</p>",
          "smithy.api#required": {}
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the license.</p>",
          "smithy.api#required": {}
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
          "smithy.api#documentation": "<p>Information about the license.</p>"
        }
      },
      "Entitlements": {
```

```
            "target": "com.amazonaws.licensemanager#EntitlementList",
            "traits": {
              "smithy.api#documentation": "<p>License entitlements.</p>",
              "smithy.api#required": {}
            }
          },
          "ConsumptionConfiguration": {
            "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
            "traits": {
              "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n        running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n        usage.</p>",
              "smithy.api#required": {}
            }
          },

   "Status": {
            "target": "com.amazonaws.licensemanager#LicenseStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>",
              "smithy.api#required": {}
            }
          },
          "ClientToken": {
            "target": "com.amazonaws.licensemanager#ClientToken",
            "traits": {
              "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the license.</p>"
            }
          }
        },
        "traits": {

 "smithy.api#input": {}
        }
      },
    "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
```

```
                    "traits": {
                        "smithy.api#documentation": "<p>License ARN.</p>"
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>New version of the license.</p>"
                    }
                },
                "Status": {
                    "target": "com.amazonaws.licensemanager#LicenseStatus",
                    "traits": {
                        "smithy.api#documentation": "<p>License status.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output":
{}
            }
        },
        "com.amazonaws.licensemanager#CreateToken": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#CreateTokenRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#CreateTokenResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RedirectException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
```

```
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n        <p>A refresh token is a JWT
token used to get an access token. With an access token,\n        you can call AssumeRoleWithWebIdentity to get
role credentials that you can use to \n        call License Manager to manage the specified license.</p>"
    }
},
"com.amazonaws.licensemanager#CreateTokenRequest": {
    "type": "structure",
    "members": {
        "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits":
{
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is
mapped to the aud claim of the\n        JWT token.</p>",
                "smithy.api#required": {}
            }
        },
        "RoleArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. \n        License Manager does not check whether the roles are in use.</p>"
            }
        },
        "ExpirationInDays": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
                "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>"
            }
        },
        "TokenProperties": {
            "target": "com.amazonaws.licensemanager#MaxSize3StringList",
            "traits": {
                "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n        to the amr claim of the JWT token.</p>"
            }
```

```
      },
      "ClientToken": {
        "target": "com.amazonaws.licensemanager#ClientToken",
        "traits": {
          "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
},
"com.amazonaws.licensemanager#CreateTokenResponse": {
  "type": "structure",
  "members": {
    "TokenId": {
      "target": "com.amazonaws.licensemanager#String",

      "traits": {
          "smithy.api#documentation": "<p>Token ID.</p>"
        }
    },
    "TokenType": {
      "target": "com.amazonaws.licensemanager#TokenType",
      "traits": {
        "smithy.api#documentation": "<p>Token type.</p>"
      }
    },
    "Token": {
      "target": "com.amazonaws.licensemanager#TokenString",
      "traits": {
        "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanager#DateTime": {
  "type": "timestamp"
},
"com.amazonaws.licensemanager#DatetimeRange": {
  "type": "structure",
  "members": {
    "Begin": {
```

```
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Start of the time range.</p>",
            "smithy.api#required": {}
          }
        },
        "End": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>End of the time range.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
    }
  },
  "com.amazonaws.licensemanager#DeleteGrant": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
        }
    },
    "com.amazonaws.licensemanager#DeleteGrantRequest": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "StatusReason": {
                "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                "traits": {
                    "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
                }
            },
            "Version": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Current version of the
grant.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#DeleteGrantResponse": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Grant ARN.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanager#GrantStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Grant status.</p>"
                }
```

```
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation":
"<p>Grant version.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicense": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"

},
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
          {
            "target": "com.amazonaws.licensemanager#ValidationException"
          }
        ],
```

```
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified license.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
      },

      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n        <p>You cannot
delete a license configuration that is in use.</p>"
      }
    },
    "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
      "type": "structure",
      "members":
{
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the license configuration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
```

```json
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n        <p>This action deletes the report generator, which stops it from generating future reports.\n        The action cannot be reversed. It has no effect on the previous reports from this generator.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
        "type": "structure",
        "members": {
          "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be
deleted.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

 "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteLicenseRequest": {
        "type": "structure",
        "members": {
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          },
          "SourceVersion": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Current version of the license.</p>",
              "smithy.api#required":
{}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

```
      },
      "com.amazonaws.licensemanager#DeleteLicenseResponse": {
        "type": "structure",
        "members": {
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
            "traits": {
              "smithy.api#documentation": "<p>License status.</p>"
            }
          },
          "DeletionDate": {
            "target": "com.amazonaws.licensemanager#ISO8601DateTime",
            "traits": {
              "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#DeleteToken": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#RedirectException"
          },
          {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          },
```

```
          {
                "target": "com.amazonaws.licensemanager#ValidationException"
          }
       ],
       "traits": {
         "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
       }
     },
     "com.amazonaws.licensemanager#DeleteTokenRequest": {
       "type": "structure",
       "members": {
         "TokenId": {
           "target": "com.amazonaws.licensemanager#String",
           "traits": {
             "smithy.api#documentation": "<p>Token ID.</p>",
             "smithy.api#required": {}
           }
         }
       },
       "traits": {
         "smithy.api#input": {}
       }
     },
     "com.amazonaws.licensemanager#DeleteTokenResponse": {
       "type": "structure",
       "members": {},
       "traits": {
         "smithy.api#output": {}

       }
     },
     "com.amazonaws.licensemanager#DigitalSignatureMethod": {
       "type": "enum",
       "members": {
         "JWT_PS384": {
           "target": "smithy.api#Unit",
           "traits": {
             "smithy.api#enumValue": "JWT_PS384"
           }
         }
       }
     },
     "com.amazonaws.licensemanager#Entitlement": {
       "type": "structure",
       "members": {
         "Name": {
```

```
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement name.</p>",
          "smithy.api#required": {}
        }
      },
      "Value": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
        }
      },
      "MaxCount": {
        "target": "com.amazonaws.licensemanager#Long",
        "traits": {
          "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
        }
      },
      "Overage": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
        }
      },
      "Unit": {
        "target": "com.amazonaws.licensemanager#EntitlementUnit",
        "traits": {
          "smithy.api#documentation": "<p>Entitlement unit.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowCheckIn": {
        "target":
"com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
    }
  },
  "com.amazonaws.licensemanager#EntitlementData": {
    "type": "structure",
    "members": {
      "Name": {
```

```
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement data name.</p>",
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement
data value.</p>"
            }
          },
          "Unit": {
            "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
            "traits": {
              "smithy.api#documentation": "<p>Entitlement data unit.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#EntitlementDataList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#EntitlementData"
        }
      },
      "com.amazonaws.licensemanager#EntitlementDataUnit": {
        "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {

"smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "SECONDS": {
```

```
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Seconds"
      }
    },
    "MICROSECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Microseconds"
      }
    },
    "MILLISECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Milliseconds"
      }
    },
    "BYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
      }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS":
```

```
{
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bits"
        }
      },
      "KILOBITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobits"
        }
      },
      "MEGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Megabits"
        }
      },
      "GIGABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Gigabits"
        }
      },
      "TERABITS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Terabits"

}
      },
      "PERCENT": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Percent"
        }
      },
      "BYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Bytes/Second"
        }
      },
      "KILOBYTES_PER_SECOND": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "Kilobytes/Second"
        }
```

```
        },
        "MEGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabytes/Second"
          }
        },
        "GIGABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabytes/Second"
          }
        },
        "TERABYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabytes/Second"
          }
        },
        "BITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bits/Second"
          }
        },
        "KILOBITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobits/Second"
          }
        },
        "MEGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue":
"Megabits/Second"
          }
        },
        "GIGABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits/Second"
          }
        },
        "TERABITS_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits/Second"
```

```
            }
          },
          "COUNT_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count/Second"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#EntitlementList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#Entitlement"
        }
      },
      "com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#EntitlementUnit": {
        "type": "enum",
        "members": {
          "COUNT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count"
            }
          },
          "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "None"
            }
          },
          "SECONDS":
{
            "target": "smithy.api#Unit",
```

```
      "traits": {
        "smithy.api#enumValue": "Seconds"
      }
    },
    "MICROSECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Microseconds"
      }
    },
    "MILLISECONDS": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Milliseconds"
      }
    },
    "BYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Bytes"
      }
    },
    "KILOBYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Kilobytes"
       }
    },
    "MEGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Megabytes"
      }
    },
    "GIGABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Gigabytes"
      }
    },
    "TERABYTES": {
      "target": "smithy.api#Unit",
      "traits": {
        "smithy.api#enumValue": "Terabytes"
      }
    },
    "BITS": {
      "target": "smithy.api#Unit",
```

```
          "traits": {
            "smithy.api#enumValue": "Bits"
          }
        },
        "KILOBITS": {
          "target": "smithy.api#Unit",
          "traits": {


            "smithy.api#enumValue": "Kilobits"
          }
        },
        "MEGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Megabits"
          }
        },
        "GIGABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Gigabits"
          }
        },
        "TERABITS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Terabits"
          }
        },
        "PERCENT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Percent"
          }
        },
        "BYTES_PER_SECOND": {
          "target":
"smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Bytes/Second"
          }
        },
        "KILOBYTES_PER_SECOND": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "Kilobytes/Second"
          }
        },
```

```
"MEGABYTES_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Megabytes/Second"
  }
},
"GIGABYTES_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Gigabytes/Second"
  }
},
"TERABYTES_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {

    "smithy.api#enumValue": "Terabytes/Second"
  }
},
"BITS_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Bits/Second"
  }
},
"KILOBITS_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Kilobits/Second"
  }
},
"MEGABITS_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Megabits/Second"
  }
},
"GIGABITS_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Gigabits/Second"
  }
},

"TERABITS_PER_SECOND": {
  "target": "smithy.api#Unit",
  "traits": {
    "smithy.api#enumValue": "Terabits/Second"
```

```
            }
          },
          "COUNT_PER_SECOND": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "Count/Second"
            }
          }
        }
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsage": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage name.</p>",
            "smithy.api#required": {}
          }
        },
        "ConsumedValue": {
          "target": "com.amazonaws.licensemanager#String",

          "traits": {
            "smithy.api#documentation": "<p>Resource usage consumed.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxCount": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
          }
        },
        "Unit": {
          "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
          "traits": {
            "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#EntitlementUsageList":
```

```json
{
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#EntitlementUsage"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>",
```

```
              "smithy.api#required": {}

          }
        },
        "DryRun": {
          "target": "com.amazonaws.licensemanager#Boolean",
          "traits": {
            "smithy.api#default": false,
            "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
      "type": "structure",
      "members": {
        "LicenseConsumptionToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License consumption token.</p>"
          }
        },


        "Expiration": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#FailedDependencyException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        },
        "ErrorCode": {
          "target": "com.amazonaws.licensemanager#String"
        }
      },
```

```
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "FailedDependency",
              "httpResponseCode": 424
            },
            "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 424
          }
        },
        "com.amazonaws.licensemanager#Filter": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "com.amazonaws.licensemanager#FilterName",
              "traits": {
                "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
              }
            },
            "Values": {
              "target": "com.amazonaws.licensemanager#FilterValues",
              "traits": {
                "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only
specify one value for the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter
 name and value pair that is used to return more specific results from a\n      describe operation. Filters can be used
to match a set of resources by specific criteria,\n      such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanager#FilterLimitExceededException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "aws.protocols#awsQueryError": {
              "code": "FilterLimitExceeded",
              "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
            "smithy.api#error": "client",
```

```
      "smithy.api#httpError": 400
    }
  },
  "com.amazonaws.licensemanager#FilterList": {
    "type": "list",
     "member": {
       "target": "com.amazonaws.licensemanager#Filter"
    }
  },
  "com.amazonaws.licensemanager#FilterName": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValue": {
    "type": "string"
  },
  "com.amazonaws.licensemanager#FilterValues": {
    "type": "list",
    "member": {
       "target": "com.amazonaws.licensemanager#FilterValue",
       "traits": {
         "smithy.api#xmlName": "item"
       }
    }
  },
  "com.amazonaws.licensemanager#Filters": {
    "type": "list",
    "member": {
       "target": "com.amazonaws.licensemanager#Filter",
       "traits": {
         "smithy.api#xmlName": "item"
       }
    }
  },
  "com.amazonaws.licensemanager#GetAccessToken": {
    "type": "operation",
    "input": {
       "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
    },
    "output": {
       "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
    },
    "errors": [
       {
         "target": "com.amazonaws.licensemanager#AccessDeniedException"
       },
       {
         "target": "com.amazonaws.licensemanager#AuthorizationException"
       },
```

```
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
 tokens\n        are valid for one hour.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetAccessTokenRequest": {
      "type": "structure",
      "members": {
        "Token": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
            "smithy.api#required": {}
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
            "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetAccessTokenResponse":
{
      "type": "structure",
      "members": {
        "AccessToken": {
          "target": "com.amazonaws.licensemanager#TokenString",
          "traits": {
            "smithy.api#documentation": "<p>Temporary access token.</p>"
          }
        }
```

```
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#GetGrant": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetGrantRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetGrantResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"

      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
    }
  },
  "com.amazonaws.licensemanager#GetGrantRequest": {
    "type": "structure",
    "members": {
      "GrantArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Grant version.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetGrantResponse": {
      "type": "structure",
      "members": {
        "Grant": {
          "target": "com.amazonaws.licensemanager#Grant",
          "traits": {
            "smithy.api#documentation": "<p>Grant details.</p>"
          }
        }
      },
      "traits": {

"smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicense": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
```

```
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"

            },
                {
                    "target": "com.amazonaws.licensemanager#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetLicenseConfiguration": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
                {
                    "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
                },
                {
                    "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets detailed information about the specified license
configuration.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
            "type": "structure",
            "members": {
                "LicenseConfigurationArn": {
```

```
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits":
{
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
        "type": "structure",
        "members": {
          "LicenseConfigurationId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
            }
          },
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
            }
          },
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the license configuration.</p>"
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the license configuration.</p>"
            }
          },
          "LicenseCountingType": {
            "target": "com.amazonaws.licensemanager#LicenseCountingType",
            "traits": {
              "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
            }
          },
          "LicenseRules": {
            "target": "com.amazonaws.licensemanager#StringList",
```

```
        "traits": {
          "smithy.api#documentation": "<p>License rules.</p>"
        }
      },
      "LicenseCount": {
        "target": "com.amazonaws.licensemanager#BoxLong",


        "traits": {
          "smithy.api#documentation": "<p>Number of available licenses.</p>"
        }
      },
      "LicenseCountHardLimit": {
        "target": "com.amazonaws.licensemanager#BoxBoolean",
        "traits": {
          "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
        }
      },
      "ConsumedLicenses": {
        "target": "com.amazonaws.licensemanager#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License configuration status.</p>"
        }


    },
      "OwnerAccountId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
        }
      },
      "ConsumedLicenseSummaryList": {
        "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
        }
      },
      "ManagedResourceSummaryList": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
        }
      },
```

```
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
           "traits": {
              "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
           }
        },
        "ProductInformationList": {
          "target": "com.amazonaws.licensemanager#ProductInformationList",
          "traits": {
              "smithy.api#documentation": "<p>Product information.</p>"
           }
        },
        "AutomatedDiscoveryInformation": {
          "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
          "traits": {
              "smithy.api#documentation": "<p>Automated discovery information.</p>"
           }
        },
        "DisassociateWhenNotFound": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
              "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
           }
        }
      }
    },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#GetLicenseConversionTask": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
```

```
                {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
                },
                {
                    "target": "com.amazonaws.licensemanager#ServerInternalException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
            }
        },
        "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
            "type": "structure",
            "members": {
                "LicenseConversionTaskId": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information
on.</p>",
                        "smithy.api#required": {}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
            "type": "structure",
            "members": {
                "LicenseConversionTaskId": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
                    "traits": {
                        "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
                    }
                },
                "ResourceArn": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license
conversion task is associated with.</p>"
                    }
                },
                "SourceLicenseContext": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                    "traits": {
                        "smithy.api#documentation": "<p>Information
```

about the license type converted from.</p>"
```
                }
              },
              "DestinationLicenseContext": {
                "target": "com.amazonaws.licensemanager#LicenseConversionContext",
                "traits": {
                  "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
                }
              },
              "StatusMessage": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
                }
              },
              "Status": {
                "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
                "traits": {
                  "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
                }
              },

          "StartTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                  "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
                }
              },
              "LicenseConversionTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                  "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
                }
              },
              "EndTime": {
                "target": "com.amazonaws.licensemanager#DateTime",
                "traits": {
                  "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
```

```
{
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
            "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"

        },
        {
            "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
            "target": "com.amazonaws.licensemanager#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
    }
},
"com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
    "type": "structure",
    "members": {
        "LicenseManagerReportGeneratorArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
                "smithy.api#required":
```

```
{}
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
            "type": "structure",
            "members": {
                "ReportGenerator": {
                    "target": "com.amazonaws.licensemanager#ReportGenerator",
                    "traits": {
                        "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseRequest": {
            "type": "structure",
            "members": {
                "LicenseArn": {
                    "target": "com.amazonaws.licensemanager#Arn",
                    "traits": {

"smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Version": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>License version.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#GetLicenseResponse": {
            "type": "structure",
            "members": {
```

```
      "License": {
        "target": "com.amazonaws.licensemanager#License",
        "traits": {
          "smithy.api#documentation": "<p>License details.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },

"com.amazonaws.licensemanager#GetLicenseUsage": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {

"target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
    }
  },
  "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
    "type": "structure",
    "members": {
```

```
          "LicenseArn": {
            "target": "com.amazonaws.licensemanager#Arn",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
        "type": "structure",
        "members": {
          "LicenseUsage": {
            "target":
"com.amazonaws.licensemanager#LicenseUsage",
            "traits": {
              "smithy.api#documentation": "<p>License usage details.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
```

```
        ],
      "traits": {
        "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "S3BucketArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n        and so on.</p>"

  }
        },
        "SnsTopicArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
          }
        },
        "OrganizationConfiguration": {
          "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n        cross-account discovery.</p>"
          }
        },
        "EnableCrossAccountsDiscovery": {
          "target": "com.amazonaws.licensemanager#BoxBoolean",
          "traits": {
            "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
          }

      },
        "LicenseManagerResourceShareArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
```

Manager management account \n   provides member accounts with access to this share.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#Grant": {
        "type": "structure",
        "members": {
            "GrantArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
                    "smithy.api#required": {}
                }
            },
            "GrantName": {

                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Grant name.</p>",
                    "smithy.api#required": {}
                }
            },
            "ParentArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>Parent ARN.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>License ARN.</p>",
                    "smithy.api#required": {}
                }
            },
            "GranteePrincipalArn": {
                "target": "com.amazonaws.licensemanager#Arn",
                "traits": {

"smithy.api#documentation": "<p>The grantee principal ARN.</p>",
                    "smithy.api#required": {}
                }
            },

```
    "HomeRegion": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Home Region of the grant.</p>",
        "smithy.api#required": {}
      }
    },
    "GrantStatus": {
      "target": "com.amazonaws.licensemanager#GrantStatus",
      "traits": {
        "smithy.api#documentation": "<p>Grant status.</p>",
        "smithy.api#required": {}
      }
    },
    "StatusReason": {
      "target": "com.amazonaws.licensemanager#StatusReasonMessage",
      "traits": {
        "smithy.api#documentation": "<p>Grant status reason.</p>"
      }
    },
    "Version": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Grant version.</p>",
        "smithy.api#required": {}
      }
    },
    "GrantedOperations": {
      "target": "com.amazonaws.licensemanager#AllowedOperationList",
      "traits": {
        "smithy.api#documentation": "<p>Granted operations.</p>",
        "smithy.api#required": {}
      }
    },
    "Options": {
      "target": "com.amazonaws.licensemanager#Options",
      "traits": {
        "smithy.api#documentation": "<p>The options specified for the grant.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>Describes a grant.</p>"
  }
},
"com.amazonaws.licensemanager#GrantList": {
  "type": "list",
```

```
      "member": {
        "target": "com.amazonaws.licensemanager#Grant"
      }
    },
    "com.amazonaws.licensemanager#GrantStatus": {
      "type": "enum",
      "members": {
        "PENDING_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_WORKFLOW"
          }
        },
        "PENDING_ACCEPT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_ACCEPT"
          }
        },
        "REJECTED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "REJECTED"

          }
        },
        "ACTIVE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "ACTIVE"
          }
        },
        "FAILED_WORKFLOW": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "FAILED_WORKFLOW"
          }
        },
        "DELETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DELETED"
          }
        },
        "PENDING_DELETE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "PENDING_DELETE"
```

```
        }
      },
      "DISABLED": {
        "target": "smithy.api#Unit",
        "traits":
{
          "smithy.api#enumValue": "DISABLED"
        }
      },
      "WORKFLOW_COMPLETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "WORKFLOW_COMPLETED"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#GrantedLicense": {
    "type": "structure",
    "members": {
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
        }
      },
      "LicenseName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>License name.</p>"
        }

  },
        "ProductName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product name.</p>"
          }
        },
        "ProductSKU": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Product SKU.</p>"
          }
        },
        "Issuer": {
          "target": "com.amazonaws.licensemanager#IssuerDetails",
          "traits": {
```

```
          "smithy.api#documentation": "<p>Granted license issuer.</p>"
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Home Region of the granted license.</p>"


        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
        "traits": {
          "smithy.api#documentation": "<p>Granted license status.</p>"
        }
      },
      "Validity": {
        "target": "com.amazonaws.licensemanager#DatetimeRange",
        "traits": {
          "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
ISO8601-UTC format.</p>"
        }
      },
      "Beneficiary": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
        }
      },
      "Entitlements": {
        "target": "com.amazonaws.licensemanager#EntitlementList",
        "traits": {
          "smithy.api#documentation": "<p>License entitlements.</p>"
        }
      },
      "ConsumptionConfiguration": {
        "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
        "traits": {
          "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
        }
      },
      "LicenseMetadata": {
        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
          "smithy.api#documentation": "<p>Granted license metadata.</p>"
        }
      },
      "CreateTime": {
```

```
                "target": "com.amazonaws.licensemanager#ISO8601DateTime",
                "traits": {
                    "smithy.api#documentation": "<p>Creation time of the granted license.</p>"

    }
        },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Version of the granted license.</p>"
            }
        },
        "ReceivedMetadata": {
            "target": "com.amazonaws.licensemanager#ReceivedMetadata",
            "traits": {
                "smithy.api#documentation": "<p>Granted license received metadata.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
    }
},
"com.amazonaws.licensemanager#GrantedLicenseList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#GrantedLicense"
    }
},
"com.amazonaws.licensemanager#ISO8601DateTime":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#pattern": "^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
    }
},
"com.amazonaws.licensemanager#Integer": {
    "type": "integer"
},
"com.amazonaws.licensemanager#InvalidParameterValueException": {
    "type": "structure",
    "members": {
        "Message": {
```

```
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidParameterValueProvided",
            "httpResponseCode": 400
          },
          "smithy.api#documentation":
  "<p>One or more parameter values are not valid.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#InvalidResourceStateException": {
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "aws.protocols#awsQueryError": {
            "code": "InvalidResourceState",
            "httpResponseCode": 400
          },
          "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n          <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },

      "com.amazonaws.licensemanager#InventoryFilter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the filter.</p>",
              "smithy.api#required": {}
            }
          },
          "Condition": {
            "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
            "traits": {
              "smithy.api#documentation": "<p>Condition of the filter.</p>",
```

```
              "smithy.api#required": {}
            }
          },
          "Value": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An inventory filter.</p>"
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterCondition": {
      "type": "enum",
      "members": {
        "EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EQUALS"
          }
        },
        "NOT_EQUALS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "NOT_EQUALS"
          }
        },
        "BEGINS_WITH": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "BEGINS_WITH"
          }
        },
        "CONTAINS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "CONTAINS"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#InventoryFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#InventoryFilter"
```

```
        }
      },
      "com.amazonaws.licensemanager#Issuer": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>",
              "smithy.api#required": {}
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
 of sign and verify, \n      and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
        }
      },
      "com.amazonaws.licensemanager#IssuerDetails": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer name.</p>"
            }
          },
          "SignKey": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n      and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
            }
          },
          "KeyFingerprint": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
            }
          }
```

```
        },
      "traits": {
        "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
      }
  },
  "com.amazonaws.licensemanager#License": {
    "type": "structure",
    "members": {
      "LicenseArn": {
        "target": "com.amazonaws.licensemanager#Arn",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
        }
      },
      "LicenseName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {

      "smithy.api#documentation": "<p>License name.</p>"
        }
      },
      "ProductName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product name.</p>"
        }
      },
      "ProductSKU": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Product SKU.</p>"
        }
      },
      "Issuer": {
        "target": "com.amazonaws.licensemanager#IssuerDetails",
        "traits": {
          "smithy.api#documentation": "<p>License issuer.</p>"
        }
      },
      "HomeRegion": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {

      "smithy.api#documentation": "<p>Home Region of the license.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanager#LicenseStatus",
```

```
          "traits": {
            "smithy.api#documentation": "<p>License status.</p>"
          }
        },
        "Validity": {
          "target": "com.amazonaws.licensemanager#DatetimeRange",
          "traits": {
            "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
          }
        },
        "Beneficiary": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License beneficiary.</p>"
          }
        },
        "Entitlements": {

    "target": "com.amazonaws.licensemanager#EntitlementList",
          "traits": {
            "smithy.api#documentation": "<p>License entitlements.</p>"
          }
        },
        "ConsumptionConfiguration": {
          "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
          "traits": {
            "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
          }
        },
        "LicenseMetadata": {
          "target": "com.amazonaws.licensemanager#MetadataList",
          "traits": {
            "smithy.api#documentation": "<p>License metadata.</p>"
          }
        },
        "CreateTime": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>License
creation time.</p>"
          }
        },
        "Version": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>License version.</p>"
          }
```

```
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfiguration": {
        "type": "structure",
        "members": {
          "LicenseConfigurationId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
            }
          },
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
            }
          },
          "Name": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Name of the license configuration.</p>"
            }
          },
          "Description": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Description of the license configuration.</p>"
            }
          },
          "LicenseCountingType": {
            "target": "com.amazonaws.licensemanager#LicenseCountingType",
            "traits": {
              "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
            }
          },
          "LicenseRules": {
            "target": "com.amazonaws.licensemanager#StringList",
            "traits": {
              "smithy.api#documentation": "<p>License rules.</p>"
            }
          },
          "LicenseCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
```

              "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
            }
          },
          "LicenseCountHardLimit": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
            "traits": {
              "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
            }
          },
          "DisassociateWhenNotFound": {
            "target": "com.amazonaws.licensemanager#BoxBoolean",
             "traits": {
              "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
            }
          },
          "ConsumedLicenses": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Status of the license configuration.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"

        }
        },
        "ConsumedLicenseSummaryList": {
          "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
          }
        },
        "ManagedResourceSummaryList": {
          "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
          }
        },
        "ProductInformationList": {

```
            "target": "com.amazonaws.licensemanager#ProductInformationList",
            "traits": {
              "smithy.api#documentation": "<p>Product information.</p>"
            }
          },
          "AutomatedDiscoveryInformation": {

            "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
            "traits": {
              "smithy.api#documentation": "<p>Automated discovery information.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license
  agreement that can be\n        consumed and enforced by License Manager. Components include specifications for
  the license\n        type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n
  Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n        must be associated with a
  host), and the number of licenses purchased and used.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
      "type": "structure",
      "members": {
        "ResourceArn":

  {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
            }
          },
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits": {
            "smithy.api#documentation": "<p>Type of server resource.</p>"
          }
        },
        "ResourceOwnerId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource
  consuming licenses.</p>"
          }
        },
        "AssociationTime": {
          "target": "com.amazonaws.licensemanager#DateTime",
          "traits": {
```

"smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
          }
        },
        "AmiAssociationScope": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-account</code>.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
      "type": "enum",
      "members":
{
        "AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISABLED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },

"smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
          }
        },
        "AmiAssociationScope": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-account</code>.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
      "type": "enum",
      "members":
{
        "AVAILABLE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "AVAILABLE"
          }
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISABLED"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },

```
            "ResourceType": {
              "target": "com.amazonaws.licensemanager#ResourceType",
              "traits":
  {
                "smithy.api#documentation": "<p>Type of resource.</p>"
              }
            },
            "ResourceStatus": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Status of the resource.</p>"
              }
            },
            "ResourceOwnerId": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
              }
            },
            "AssociationTime": {
              "target": "com.amazonaws.licensemanager#DateTime",
              "traits": {
                "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
              }
            },

      "ConsumedLicenses": {
              "target": "com.amazonaws.licensemanager#BoxLong",
              "traits": {
                "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
        }
      },
      "com.amazonaws.licensemanager#LicenseConfigurations": {
        "type": "list",
        "member": {
```

```
            "target": "com.amazonaws.licensemanager#LicenseConfiguration"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionContext": {

  "type": "structure",
        "members": {
            "UsageOperation": {
                "target": "com.amazonaws.licensemanager#UsageOperation",
                "traits": {
                    "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from.  For more information about which platforms correspond to which usage
operation values see <a href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\n      </a>\n        </p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
    },
    "com.amazonaws.licensemanager#LicenseConversionTask": {
        "type": "structure",
        "members": {
            "LicenseConversionTaskId": {

    "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
            }
        },
        "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\n        conversion task.</p>"
            }
        },
        "SourceLicenseContext": {
            "target": "com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
                "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
            }
        },
        "DestinationLicenseContext": {
            "target":
```

"com.amazonaws.licensemanager#LicenseConversionContext",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the conversion task.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
            }
          },
          "StartTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation":
 "<p>The time the conversion task was started at.</p>"
            }
          },
          "LicenseConversionTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
            }
          },
          "EndTime": {
            "target": "com.amazonaws.licensemanager#DateTime",
            "traits": {
              "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
        }
      },
      "com.amazonaws.licensemanager#LicenseConversionTaskId": {
        "type": "string",
        "traits": {

```
    "smithy.api#length": {
          "min": 0,
          "max": 50
      },
      "smithy.api#pattern": "^lct-[a-zA-Z0-9]*$"
    }
  },
  "com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
    "type": "enum",
    "members": {
      "IN_PROGRESS": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "IN_PROGRESS"
        }
      },
      "SUCCEEDED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "SUCCEEDED"
        }
      },
      "FAILED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "FAILED"
        }
      }
    }
  },
  "com.amazonaws.licensemanager#LicenseConversionTasks":
{
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanager#LicenseConversionTask"
    }
  },
  "com.amazonaws.licensemanager#LicenseCountingType": {
    "type": "enum",
    "members": {
      "VCPU": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "vCPU"
        }
      },
      "INSTANCE": {
        "target": "smithy.api#Unit",
```

```
            "traits": {
                "smithy.api#enumValue": "Instance"
            }
        },
        "CORE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Core"
            }
        },
        "SOCKET": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Socket"
            }
        }
    }
},
"com.amazonaws.licensemanager#LicenseDeletionStatus": {
    "type": "enum",
    "members": {
        "PENDING_DELETE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "PENDING_DELETE"
            }
        },
        "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "DELETED"
            }
        }
    }
},
"com.amazonaws.licensemanager#LicenseList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanager#License"
    }
},
"com.amazonaws.licensemanager#LicenseOperationFailure": {

    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
```

```
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
        }
      },
      "ResourceType": {
        "target": "com.amazonaws.licensemanager#ResourceType",
        "traits": {
          "smithy.api#documentation": "<p>Resource type.</p>"
        }
      },
      "ErrorMessage": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Error message.</p>"
        }
      },
      "FailureTime": {
        "target": "com.amazonaws.licensemanager#DateTime",
        "traits": {

    "smithy.api#documentation": "<p>Failure time.</p>"
        }
      },
      "OperationName": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Name of the operation.</p>"
        }
      },
      "ResourceOwnerId": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
        }
      },
      "OperationRequestedBy": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
        }
      },
      "MetadataList": {

        "target": "com.amazonaws.licensemanager#MetadataList",
        "traits": {
          "smithy.api#documentation": "<p>Reserved.</p>"
        }
      }
```

```
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
          }
        },
        "com.amazonaws.licensemanager#LicenseOperationFailureList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
          }
        },
        "com.amazonaws.licensemanager#LicenseSpecification": {
          "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

                "smithy.api#required": {}
              }
            },
            "AmiAssociationScope": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
          }
        },
        "com.amazonaws.licensemanager#LicenseSpecifications": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#LicenseSpecification"
          }
        },
        "com.amazonaws.licensemanager#LicenseStatus": {
          "type": "enum",
          "members": {
            "AVAILABLE": {
              "target": "smithy.api#Unit",
```

```
      "traits": {
          "smithy.api#enumValue": "AVAILABLE"
        }
      },
      "PENDING_AVAILABLE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_AVAILABLE"
        }
      },
      "DEACTIVATED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DEACTIVATED"
        }
      },
      "SUSPENDED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "SUSPENDED"
        }
      },
      "EXPIRED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "EXPIRED"
        }
      },

    "PENDING_DELETE": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "PENDING_DELETE"
        }
      },
      "DELETED": {
        "target": "smithy.api#Unit",
        "traits": {
          "smithy.api#enumValue": "DELETED"
        }
      }
    }
},
"com.amazonaws.licensemanager#LicenseUsage": {
  "type": "structure",
  "members": {
    "EntitlementUsages": {
      "target": "com.amazonaws.licensemanager#EntitlementUsageList",
```

```
          "traits": {
              "smithy.api#documentation": "<p>License entitlement usages.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"

      }
    },
    "com.amazonaws.licensemanager#LicenseUsageException": {
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "aws.protocols#awsQueryError": {
          "code": "LicenseUsageFailure",
          "httpResponseCode": 412
        },
        "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 412
      }
    },
    "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
      },
      "output": {

    "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
```

```
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the resource associations for the
 specified license configuration.</p>\n        <p>Resource associations need not consume licenses from a license
configuration. \n        For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
        "type": "structure",
        "members": {
          "LicenseConfigurationArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
        "type": "structure",
        "members": {
          "LicenseConfigurationAssociations": {
```

```
                    "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
                    "traits": {
                        "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
                    }
                },
                "NextToken": {

 "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListDistributedGrants": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
```

```
            ],
            "traits": {
                "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
            }
        },
        "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
            "type": "structure",
            "members": {
                "GrantArns": {
                    "target": "com.amazonaws.licensemanager#ArnList",
                    "traits": {

                        "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
                    }
                },
                "Filters": {
                    "target": "com.amazonaws.licensemanager#FilterList",
                    "traits": {
                        "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n        <p>\n                    <code>LicenseArn</code>\n        </p>\n
        </li>\n        <li>\n        <p>\n                    <code>GrantStatus</code>\n        </p>\n        </li>\n
    <li>\n        <p>\n                    <code>GranteePrincipalARN</code>\n        </p>\n        </li>\n
<li>\n        <p>\n                    <code>ProductSKU</code>\n        </p>\n        </li>\n        <li>\n
<p>\n                    <code>LicenseIssuerName</code>\n
    </p>\n        </li>\n        </ul>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                },
                "MaxResults": {
                    "target": "com.amazonaws.licensemanager#MaxSize100",
                    "traits": {
                        "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
            "type": "structure",
            "members": {
                "Grants": {
```

```
            "target": "com.amazonaws.licensemanager#GrantList",

      "traits": {
                "smithy.api#documentation": "<p>Distributed grant details.</p>"
              }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
      },
      "errors": [
        {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
```

```
            "type": "structure",
          "members": {
            "LicenseConfigurationArn": {
              "target": "com.amazonaws.licensemanager#String",
               "traits": {
                 "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
                 "smithy.api#required": {}
               }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      }
    },
    "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
{
        "type": "structure",
        "members": {
          "LicenseOperationFailureList": {
            "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
            "traits": {
              "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurations": {
```

```
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
      },

  "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the license configurations
 for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArns": {
          "target": "com.amazonaws.licensemanager#StringList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
```

          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next

set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#Filters",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical

operators\n   are supported:</p>\n   \n   \n   <p>\n
<code>licenseCountingType</code> - The dimension for which licenses are counted.\n   Possible values are
<code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n   Logical
operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   \n
<p>\n   <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license

enforcement is used. \n   Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p>\n

  \n   \n   <p>\n   <code>usagelimitExceeded</code> - A Boolean value that
indicates whether the available licenses have been exceeded. \n   Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n   \n   "
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
        "type": "structure",
        "members": {
          "LicenseConfigurations": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurations",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",

          "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}

```
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
        "type": "structure",
        "members": {
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {

"smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "Filters": {
```

```
                    "target": "com.amazonaws.licensemanager#Filters",
                    "traits": {
                        "smithy.api#documentation": "<p>\n        Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n      </p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
            "type": "structure",
            "members": {
                "LicenseConversionTasks": {
                    "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
                    "traits": {
                        "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
                    }
                },
                "NextToken": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target":
"com.amazonaws.licensemanager#AuthorizationException"
                },
```

```
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n    <ul>\n      <li>\n        <p>\n          <code>LicenseConfigurationArn</code>\n
</p>\n      </li>\n    </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits":
{
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
```

```
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
      "type": "structure",
      "members": {
        "ReportGenerators": {
          "target": "com.amazonaws.licensemanager#ReportGeneratorList",
          "traits": {
            "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license
configurations.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }

        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
```

```
          },
          {
            "target": "com.amazonaws.licensemanager#ServerInternalException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {

      "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
        "type": "structure",
        "members": {
          "LicenseSpecifications": {
            "target": "com.amazonaws.licensemanager#LicenseSpecifications",
            "traits": {
              "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
            }
          },
          "NextToken": {
```

```
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanager#ListLicenseVersions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanager#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanager#AuthorizationException"
    },
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
  }
},
"com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
  "type": "structure",
  "members": {
    "LicenseArn": {
      "target": "com.amazonaws.licensemanager#Arn",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
        "smithy.api#required": {}
      }
```

```
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",
              "traits": {
                "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
          "type": "structure",
          "members": {
            "Licenses": {
              "target": "com.amazonaws.licensemanager#LicenseList",
              "traits": {
                "smithy.api#documentation": "<p>License details.</p>"
              }
            },

"NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanager#ListLicenses": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanager#ListLicensesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanager#ListLicensesResponse"
          },
          "errors": [
```

```
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
         },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListLicensesRequest": {
      "type": "structure",
      "members": {
        "LicenseArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
          }
        },

    "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n        <p>\n        <code>Beneficiary</code>\n        </p>\n
      </li>\n        <li>\n        <p>\n        <code>ProductSKU</code>\n        </p>\n        </li>\n
  <li>\n        <p>\n        <code>Fingerprint</code>\n        </p>\n        </li>\n        <li>\n
<p>\n        <code>Status</code>\n        </p>\n        </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token
```

for the next set of results.</p>"
```
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                  "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#ListLicensesResponse": {
            "type": "structure",
            "members": {
              "Licenses": {
                "target": "com.amazonaws.licensemanager#LicenseList",
                "traits": {
                  "smithy.api#documentation": "<p>License details.</p>"
                }
              },
              "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {

  "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#ListReceivedGrants": {
            "type": "operation",
            "input": {
              "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
            },
            "output": {
              "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
            },
            "errors": [
              {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
              },
              {
```

```
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while
specifying the\n        recipient as this Amazon Web Services account, your organization, or an organizational unit\n
    (OU) to which this member account belongs.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
```

```
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
      "type": "structure",
      "members": {
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
            "smithy.api#required": {}
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
    <ul>\n        <li>\n          <p>\n            <code>ParentArn</code>\n          </p>\n        </li>\n
<li>\n          <p>\n            <code>GranteePrincipalArn</code>\n          </p>\n        </li>\n      </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
```

"com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
    "type": "structure",
    "members": {
      "Grants": {
        "target": "com.amazonaws.licensemanager#GrantList",
        "traits": {
          "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
    "type": "structure",
    "members": {
      "GrantArns": {
        "target": "com.amazonaws.licensemanager#ArnList",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanager#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:</p>\n \n \n <p>\n <code>ProductSKU</code>\n </p>\n \n \n <p>\n <code>LicenseIssuerName</code>\n </p>\n \n \n <p>\n <code>LicenseArn</code>\n </p>\n \n \n <p>\n <code>GrantStatus</code>\n </p>\n \n \n <p>\n <code>GranterAccountId</code>\n </p>\n \n "
        }
      },
      "NextToken": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Token for the next set of results.</p>"
        }
      },

```
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
      "type": "structure",
      "members": {
        "Grants": {
          "target": "com.amazonaws.licensemanager#GrantList",
          "traits": {
            "smithy.api#documentation": "<p>Received grant details.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicenses": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
```

```
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
        "target": "com.amazonaws.licensemanager#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists received licenses.</p>"
    }
  },
  "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
  {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
      },
      {

        "target": "com.amazonaws.licensemanager#ServerInternalException"
      },
      {
```

```
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanager#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n        <ul>\n        <li>\n        <p>\n            <code>Beneficiary</code>\n        </p>\n
        </li>\n        <li>\n        <p>\n            <code>ProductSKU</code>\n
        </p>\n        </li>\n        </ul>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#MaxSize100",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
      "type": "structure",
      "members": {
        "Licenses": {
          "target": "com.amazonaws.licensemanager#GrantedLicenseList",
          "traits": {
            "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
```

                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
        "type": "structure",
        "members": {
            "LicenseArns": {
                "target": "com.amazonaws.licensemanager#ArnList",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
                }
            },

            "Filters": {
                "target": "com.amazonaws.licensemanager#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n   \n   \n   <p>\n   <code>ProductSKU</code>\n   </p>\n
        \n   \n   <p>\n   <code>Status</code>\n   </p>\n   \n
\n   <p>\n   <code>Fingerprint</code>\n   </p>\n   \n   \n
<p>\n   <code>IssuerName</code>\n   </p>\n   \n   \n   <p>\n
<code>Beneficiary</code>\n   </p>\n   \n   "
                }
            },
            "NextToken": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Token for the next set of results.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanager#MaxSize100",
                "traits": {
                    "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
        "type": "structure",

```
        "members": {
          "Licenses": {
            "target": "com.amazonaws.licensemanager#GrantedLicenseList",
            "traits": {
              "smithy.api#documentation": "<p>Received license details.</p>"
            }
          },

          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListResourceInventory": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#FailedDependencyException"
        },
        {
          "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
```

```
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
        }
      },
      "com.amazonaws.licensemanager#ListResourceInventoryRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanager#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanager#InventoryFilterList",
            "traits": {
              "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n          are supported:</p>\n          <ul>\n            <li>\n              <p>\n                <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
            </li>\n            <li>\n              <p>\n                <code>application_name</code> - The name of the
application. \n            Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
          </li>\n            <li>\n              <p>\n                <code>license_included</code> - The type of license included. \n
          Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n            Possible values
are <code>sql-server-enterprise</code> | \n            <code>sql-server-standard</code> | \n            <code>sql-
server-web</code> | \n            <code>windows-server-datacenter</code>.</p>\n            </li>\n            <li>\n
  <p>\n              <code>platform</code> - The platform of the resource.\n            Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n            </li>\n            <li>\n              <p>\n
      <code>resource_id</code> - The ID of the resource. \n            Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n            </li>\n            <li>\n              <p>\n
  <code>tag:<key></code> - The key/value combination of a tag assigned \n            to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n            <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n            </li>\n          </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

```
    "com.amazonaws.licensemanager#ListResourceInventoryResponse": {
      "type": "structure",
      "members": {
        "ResourceInventoryList": {
          "target": "com.amazonaws.licensemanager#ResourceInventoryList",
          "traits": {
            "smithy.api#documentation": "<p>Information
about the resources.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
```

```json
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanager#TagList",
          "traits": {
            "smithy.api#documentation": "<p>Information about the tags.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#ListTokens": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#ListTokensRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#ListTokensResponse"
      },
      "errors":
[
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
              "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
            {
              "target": "com.amazonaws.licensemanager#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists your tokens.</p>"
          }
        },
        "com.amazonaws.licensemanager#ListTokensRequest": {
          "type": "structure",
          "members": {
            "TokenIds": {
              "target": "com.amazonaws.licensemanager#StringList",
              "traits": {

                "smithy.api#documentation": "<p>Token IDs.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanager#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
    <ul>\n        <li>\n          <p>\n            <code>LicenseArns</code>\n          </p>\n        </li>\n
</ul>"
              }
            },
            "NextToken": {
              "target": "com.amazonaws.licensemanager#String",
              "traits": {
                "smithy.api#documentation": "<p>Token for the next set of results.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanager#MaxSize100",
              "traits": {
                "smithy.api#documentation":
"<p>Maximum number of results to return in a single call.</p>"
              }
            }
          },
          "traits": {
```

```
          "smithy.api#input": { }
        }
      },
      "com.amazonaws.licensemanager#ListTokensResponse": {
        "type": "structure",
        "members": {
          "Tokens": {
            "target": "com.amazonaws.licensemanager#TokenList",
            "traits": {
              "smithy.api#documentation": "<p>Received token details.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": { }
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
{
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanager#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanager#AuthorizationException"
          },
          {
            "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
          },
          {
            "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
          },
          {
            "target": "com.amazonaws.licensemanager#RateLimitExceededException"
          },
```

```
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n        consumption details by resource at a selected point in time. Use this action to audit the\n        current
license consumption for any license inventory and configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanager#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
          }
        },
        "NextToken": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token for the next set of results.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanager#Filters",
          "traits": {
            "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n        are supported:</p>\n        <ul>\n        <li>\n        <p>\n
<code>resourceArn</code> - The ARN of the license configuration resource.\n
        Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n
<li>\n        <p>\n                <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n        Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        <li>\n        <p>\n
    <code>resourceAccount</code> - The ID of the account that owns the resource. \n        Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n        </li>\n        </ul>"
          }
        }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
        "type": "structure",
        "members": {

  "LicenseConfigurationUsageList": {
            "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
            "traits": {
              "smithy.api#documentation": "<p>Information about the license configurations.</p>"
            }
          },
          "NextToken": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Token for the next set of results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanager#Location": {
        "type": "string"
      },
      "com.amazonaws.licensemanager#Long": {
        "type": "long"
      },
      "com.amazonaws.licensemanager#ManagedResourceSummary": {
        "type": "structure",
        "members": {
          "ResourceType":
{
            "target": "com.amazonaws.licensemanager#ResourceType",
            "traits": {
              "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
            }
          },
          "AssociationCount": {
            "target": "com.amazonaws.licensemanager#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
            }
          }
```

```
      },
      "traits": {
        "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
      }
    },
    "com.amazonaws.licensemanager#MaxSize100": {

      "type": "integer",
      "traits": {
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "com.amazonaws.licensemanager#MaxSize3StringList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 3
        }
      }
    },
    "com.amazonaws.licensemanager#Message": {
      "type": "string"
    },
    "com.amazonaws.licensemanager#Metadata": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>The key name.</p>"
          }
        },

        "Value": {
          "target": "com.amazonaws.licensemanager#String",
```

```
              "traits": {
                "smithy.api#documentation": "<p>The value.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
          }
        },
        "com.amazonaws.licensemanager#MetadataList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanager#Metadata"
          }
        },
        "com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
          "type": "structure",
          "members": {
            "Message": {
              "target": "com.amazonaws.licensemanager#Message"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
          }
        },
        "com.amazonaws.licensemanager#Options": {
          "type": "structure",
          "members": {
            "ActivationOverrideBehavior": {
              "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
              "traits": {
                "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n        Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n        <ul>\n          <li>\n            <p>As a license administrator, you can
optionally specify an\n            <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
  </li>\n          <li>\n            <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n            account in your
organization.</p>\n          </li>\n          <li>\n            <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n            <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n          are activating the grant.</p>\n          </li>\n        </ul>\n        <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n          <dd>\n            <p>Use this value to activate a grant
without replacing any member accounts\n            active grants for the same product.</p>\n          </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n          <dd>\n            <p>Use this value to activate a
```

grant and disable other active grants in any\n   member accounts
 for the same product. This action will also replace their\n   previously activated grants with this activated
grant.</p>\n   </dd>\n   </dl>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n   override behavior. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager in the <i>License Manager User Guide</i>.</p>"
        }
    },
    "com.amazonaws.licensemanager#OrganizationConfiguration": {
        "type": "structure",
        "members": {
            "EnableIntegration": {
                "target": "com.amazonaws.licensemanager#Boolean",
                "traits": {
                    "smithy.api#default": false,
                    "smithy.api#documentation":
 "<p>Enables Organizations integration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
        }
    },
    "com.amazonaws.licensemanager#PrincipalArnList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#Arn"
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 1
            }
        }
    },
    "com.amazonaws.licensemanager#ProductInformation": {
        "type": "structure",
        "members": {
            "ResourceType": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {

```
                "smithy.api#documentation": "<p>Resource type. The
possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
                "smithy.api#required": {}
            }
        },
        "ProductInformationFilterList": {
            "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
            "traits": {
                "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n        <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>. </p>\n        <p>The following filters and are supported when
the resource type \n        is <code>SSM_MANAGED</code>:</p>\n
        <ul>\n            <li>\n                <p>\n                    <code>Application Name</code> - The name of the
application.\n          Logical operator is <code>EQUALS</code>.</p>\n            </li>\n            <li>\n
<p>\n                    <code>Application Publisher</code> - The publisher of the application.\n          Logical
operator is <code>EQUALS</code>.</p>\n            </li>\n            <li>\n                <p>\n
<code>Application Version</code> - The version of the application.\n          Logical operator is
<code>EQUALS</code>.</p>\n            </li>\n            <li>\n                <p>\n                    <code>Platform
Name</code> - The name of the platform.\n          Logical operator is <code>EQUALS</code>.</p>\n
</li>\n            <li>\n                <p>\n                    <code>Platform Type</code> - The platform type.\n
Logical operator is <code>EQUALS</code>.</p>\n
        </li>\n            <li>\n                <p>\n                    <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>.  The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n          Logical operator is <code>NOT_EQUALS</code>.</p>\n            </li>\n
<li>\n                <p>\n                    <code>License Included</code> - The type of license included.\n          Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n          Possible values are:
 <code>sql-server-enterprise</code> | \n                   <code>sql-server-standard</code> | \n                    <code>sql-server-
web</code> | \n                    <code>windows-server-datacenter</code>.</p>\n            </li>\n        </ul>\n
<p>The following filters and logical operators are supported when the resource type\n        is
<code>RDS</code>:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>Engine Edition</code> -
The edition of the database engine.\n          Logical operator is <code>EQUALS</code>.\n          Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n            </li>\n            <li>\n                <p>\n                    <code>License Pack</code> - The license
pack.\n          Logical operator is <code>EQUALS</code>.\n          Possible values are: <code>data
guard</code> | \n                    <code>diagnostic pack sqlt</code> | \n
        <code>tuning pack sqlt</code> | \n                    <code>ols</code> | \n                    <code>olap</code>.</p>\n
    </li>\n        </ul>",
                "smithy.api#required": {}
            }
        }
    },
```

```json
      "traits": {
        "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilter": {
      "type": "structure",
      "members": {
        "ProductInformationFilterName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Filter name.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductInformationFilterValue": {
          "target": "com.amazonaws.licensemanager#StringList",

        "traits": {
            "smithy.api#documentation": "<p>Filter value.</p>"
          }
        },
        "ProductInformationFilterComparator": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Logical operator.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes product information filters.</p>"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationFilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformationFilter"
      }
    },
    "com.amazonaws.licensemanager#ProductInformationList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ProductInformation"
      }
    },
    "com.amazonaws.licensemanager#ProvisionalConfiguration": {
      "type": "structure",
      "members": {
```

```
                "MaxTimeToLiveInMinutes": {
                  "target": "com.amazonaws.licensemanager#BoxInteger",
                  "traits": {
                    "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
                    "smithy.api#required": {}
                  }
                }
              },
              "traits": {
                "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
              }
            },
            "com.amazonaws.licensemanager#RateLimitExceededException": {
              "type": "structure",
              "members": {
                "Message": {
                  "target": "com.amazonaws.licensemanager#Message"
                }
              },
              "traits": {
                "aws.protocols#awsQueryError": {
                  "code": "RateLimitExceeded",
                  "httpResponseCode": 429
                },
                "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
                "smithy.api#error": "client",
                "smithy.api#httpError": 429
              }
            },
            "com.amazonaws.licensemanager#ReceivedMetadata": {
              "type": "structure",
              "members": {
                "ReceivedStatus": {
                  "target": "com.amazonaws.licensemanager#ReceivedStatus",
                  "traits": {
                    "smithy.api#documentation": "<p>Received status.</p>"
                  }
                },
                "ReceivedStatusReason": {
                  "target": "com.amazonaws.licensemanager#StatusReasonMessage",
                  "traits": {
                    "smithy.api#documentation": "<p>Received
status reason.</p>"
                  }
                },
                "AllowedOperations": {
                  "target": "com.amazonaws.licensemanager#AllowedOperationList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Allowed operations.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
        }
      },
      "com.amazonaws.licensemanager#ReceivedStatus": {
        "type": "enum",
        "members": {
          "PENDING_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "PENDING_WORKFLOW"
            }
          },
          "PENDING_ACCEPT": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue":
"PENDING_ACCEPT"
            }
          },
          "REJECTED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "REJECTED"
            }
          },
          "ACTIVE": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "ACTIVE"
            }
          },
          "FAILED_WORKFLOW": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "FAILED_WORKFLOW"
            }
          },
          "DELETED": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DELETED"
            }
```

```
        },
        "DISABLED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "DISABLED"
          }
        },
        "WORKFLOW_COMPLETED": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "WORKFLOW_COMPLETED"
          }
        }
      }
    }
  },
  "com.amazonaws.licensemanager#RedirectException": {
    "type": "structure",
    "members": {
      "Location": {
        "target": "com.amazonaws.licensemanager#Location",
        "traits": {
          "smithy.api#httpHeader": "Location"
        }
      },
      "Message": {
        "target": "com.amazonaws.licensemanager#Message"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 308
    }
  },
  "com.amazonaws.licensemanager#RejectGrant": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#RejectGrantRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#RejectGrantResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      }
```

```
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        },
        {
          "target": "com.amazonaws.licensemanager#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
      }
    },
    "com.amazonaws.licensemanager#RejectGrantRequest": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}

      }
    },
    "com.amazonaws.licensemanager#RejectGrantResponse": {
      "type": "structure",
      "members": {
        "GrantArn": {
          "target": "com.amazonaws.licensemanager#Arn",
          "traits": {
            "smithy.api#documentation": "<p>Grant ARN.</p>"
          }
        },
        "Status": {
```

```
            "target": "com.amazonaws.licensemanager#GrantStatus",
            "traits": {
                "smithy.api#documentation": "<p>Grant status.</p>"
            }
        },
        "Version": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
                "smithy.api#documentation": "<p>Grant version.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#RenewType":
{
    "type": "enum",
    "members": {
        "NONE": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "None"
            }
        },
        "WEEKLY": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Weekly"
            }
        },
        "MONTHLY": {
            "target": "smithy.api#Unit",
            "traits": {
                "smithy.api#enumValue": "Monthly"
            }
        }
    }
},
"com.amazonaws.licensemanager#ReportContext": {
    "type": "structure",
    "members": {
        "licenseConfigurationArns": {
            "target": "com.amazonaws.licensemanager#ArnList",
            "traits": {
```

```
                "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
                  "smithy.api#required": {}
              }
            }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportFrequency": {
        "type": "structure",
        "members": {
          "value": {
            "target": "com.amazonaws.licensemanager#Integer",
            "traits": {
              "smithy.api#documentation": "<p>Number of times within the frequency period that a report is
generated. \n       The only supported value is <code>1</code>.</p>"
            }
          },
          "period": {
            "target": "com.amazonaws.licensemanager#ReportFrequencyType",

            "traits": {
              "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
        }
    },
    "com.amazonaws.licensemanager#ReportFrequencyType": {
        "type": "enum",
        "members": {
          "DAY": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "DAY"
            }
          },
          "WEEK": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "WEEK"
            }
          },
```

```
        "MONTH": {
          "target": "smithy.api#Unit",
          "traits":

{

            "smithy.api#enumValue": "MONTH"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportGenerator": {
      "type": "structure",
      "members": {
        "ReportGeneratorName": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the report generator.</p>"
          }
        },
        "ReportType": {
          "target": "com.amazonaws.licensemanager#ReportTypeList",
          "traits": {
            "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
          }
        },
        "ReportContext": {
          "target": "com.amazonaws.licensemanager#ReportContext",
          "traits": {
            "smithy.api#documentation": "<p>License

configuration type for this generator.</p>"
          }
        },
        "ReportFrequency": {
          "target": "com.amazonaws.licensemanager#ReportFrequency",
          "traits": {
            "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
          }
        },
        "LicenseManagerReportGeneratorArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
          }
        },
        "LastRunStatus": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
          }
```

```
            },
        "LastRunFailureReason":
{
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
            }
        },
        "LastReportGenerationTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
            }
        },
        "ReportCreatorAccount": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
            }
        },
        "Description": {
            "target": "com.amazonaws.licensemanager#String",

        "traits": {
              "smithy.api#documentation": "<p>Description of the report generator.</p>"
            }
        },
        "S3Location": {
            "target": "com.amazonaws.licensemanager#S3Location",
            "traits": {
              "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
            }
        },
        "CreateTime": {
            "target": "com.amazonaws.licensemanager#String",
            "traits": {
              "smithy.api#documentation": "<p>Time the report was created.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanager#TagList",
            "traits": {
              "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
            }
```

```
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportGenerator"
      }
    },
    "com.amazonaws.licensemanager#ReportGeneratorName": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "com.amazonaws.licensemanager#ReportType": {
      "type": "enum",
      "members": {
        "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
          }

        },
        "LICENSE_CONFIGURATION_USAGE_REPORT": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "LicenseConfigurationUsageReport"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#ReportTypeList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ReportType"
      }
    },
    "com.amazonaws.licensemanager#ResourceInventory": {
      "type": "structure",
      "members": {
```

```
        "ResourceId": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the resource.</p>"
          }
        },
        "ResourceType": {
          "target": "com.amazonaws.licensemanager#ResourceType",
          "traits":

{

            "smithy.api#documentation": "<p>Type of resource.</p>"
          }
        },
        "ResourceArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
          }
        },
        "Platform": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Platform of the resource.</p>"
          }
        },
        "PlatformVersion": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
          }
        },
        "ResourceOwningAccountId":

{

          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a resource.</p>"
      }
    },
    "com.amazonaws.licensemanager#ResourceInventoryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#ResourceInventory"
      }
```

```
    },
    "com.amazonaws.licensemanager#ResourceLimitExceededException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "ResourceLimitExceeded",

                "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "Message": {
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "InvalidResource.NotFound",
                "httpResponseCode": 400
            },
            "smithy.api#documentation": "<p>The resource cannot be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 400
        }
    },
    "com.amazonaws.licensemanager#ResourceType": {
        "type": "enum",
        "members": {
            "EC2_INSTANCE": {
                "target": "smithy.api#Unit",
                "traits": {
                    "smithy.api#enumValue": "EC2_INSTANCE"
                }
            },
            "EC2_HOST": {
                "target": "smithy.api#Unit",
                "traits": {
```

```
            "smithy.api#enumValue": "EC2_HOST"
          }
        },
        "EC2_AMI": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "EC2_AMI"
          }
        },
        "RDS": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "RDS"
          }
        },
        "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
          "target": "smithy.api#Unit",
          "traits": {
            "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
          }
        }
      }
    },
    "com.amazonaws.licensemanager#S3Location": {
      "type": "structure",
      "members": {
        "bucket": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
          }
        },
        "keyPrefix": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
      }
    },
    "com.amazonaws.licensemanager#ServerInternalException": {
      "type": "structure",
      "members": {
        "Message": {
```

```
                "target": "com.amazonaws.licensemanager#Message"
            }
        },
        "traits": {
            "aws.protocols#awsQueryError": {
                "code": "InternalError",
                "httpResponseCode": 500
            },
            "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanager#SignedToken": {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 4096
            }
        }
    },
    "com.amazonaws.licensemanager#StatusReasonMessage": {
        "type": "string",
        "traits":
{
            "smithy.api#length": {
                "min": 0,
                "max": 400
            },
            "smithy.api#pattern": "^[\\s\\S]+$"
        }
    },
    "com.amazonaws.licensemanager#String": {
        "type": "string"
    },
    "com.amazonaws.licensemanager#StringList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanager#String"
        }
    },
    "com.amazonaws.licensemanager#Tag": {
        "type": "structure",
        "members": {
            "Key": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Tag key.</p>"
```

```
          }
        },
        "Value": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation":
"<p>Tag value.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#TagKeyList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#String"
      }
    },
    "com.amazonaws.licensemanager#TagList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanager#Tag"
      }
    },
    "com.amazonaws.licensemanager#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanager#TagResourceResponse"
      },
      "errors": [
        {
          "target":
"com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
```

```
              {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
            }
          },
          "com.amazonaws.licensemanager#TagResourceRequest": {
            "type": "structure",
            "members": {
              "ResourceArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits":
  {
                  "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
                  "smithy.api#required": {}
                }
              },
              "Tags": {
                "target": "com.amazonaws.licensemanager#TagList",
                "traits": {
                  "smithy.api#documentation": "<p>One or more tags.</p>",
                  "smithy.api#required": {}
                }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanager#TagResourceResponse": {
            "type": "structure",
            "members": {},
            "traits": {
              "smithy.api#output": {}
            }
          },
          "com.amazonaws.licensemanager#TokenData": {
            "type": "structure",
            "members": {
              "TokenId": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                  "smithy.api#documentation": "<p>Token ID.</p>"
                }
              },
```

```
        "TokenType": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
          }
        },
        "LicenseArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
          }
        },
        "ExpirationTime": {
          "target": "com.amazonaws.licensemanager#ISO8601DateTime",
          "traits": {
            "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
 format.</p>"
          }
        },
        "TokenProperties": {
          "target": "com.amazonaws.licensemanager#MaxSize3StringList",
          "traits": {
            "smithy.api#documentation": "<p>Data specified by the caller.</p>"
          }
        },
        "RoleArns": {
          "target": "com.amazonaws.licensemanager#ArnList",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
          }
        }
      },
      "traits": {

        "smithy.api#documentation": "<p>Describes a token.</p>"
      }
    },
    "com.amazonaws.licensemanager#TokenList": {
      "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanager#TokenData"
        }
      },
      "com.amazonaws.licensemanager#TokenString": {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 4096
          },
          "smithy.api#pattern": "^\\S+$"
        }
      },
      "com.amazonaws.licensemanager#TokenType": {
        "type": "enum",
        "members": {
          "REFRESH_TOKEN": {
            "target": "smithy.api#Unit",
            "traits": {
              "smithy.api#enumValue": "REFRESH_TOKEN"
            }
          }
        }
      },
      "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
{
        "type": "structure",
        "members": {
          "Message": {
            "target": "com.amazonaws.licensemanager#Message"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request
again.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 400
        }
      },
      "com.amazonaws.licensemanager#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanager#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanager#UntagResourceResponse"
        },
```

```
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
       },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
    }
  },
  "com.amazonaws.licensemanager#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
          "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
          "smithy.api#required": {}
        }
      },
      "TagKeys": {
        "target": "com.amazonaws.licensemanager#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanager#UntagResourceResponse": {
    "type": "structure",
    "members": {},
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
      },
      "output": {

"target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanager#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanager#AuthorizationException"
        },
        {
          "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
        },
        {
          "target": "com.amazonaws.licensemanager#RateLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
        },
        {
          "target": "com.amazonaws.licensemanager#ServerInternalException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
      }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
      "type": "structure",
      "members": {
        "LicenseConfigurationArn": {
          "target": "com.amazonaws.licensemanager#String",
          "traits": {
            "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
            "smithy.api#required": {}
          }
        },
```

```
"LicenseConfigurationStatus": {
    "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
    "traits": {
        "smithy.api#documentation": "<p>New status of the license configuration.</p>"
    }
},
"LicenseRules": {
    "target": "com.amazonaws.licensemanager#StringList",
    "traits": {

        "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\n
    configuration is licenseAffinityToHost.</p>"
    }
},
"LicenseCount": {
    "target": "com.amazonaws.licensemanager#BoxLong",
    "traits": {
        "smithy.api#documentation": "<p>New number of licenses managed by the license
configuration.</p>"
    }
},
"LicenseCountHardLimit": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
    "traits": {
        "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
    }
},
"Name": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>New name
of the license configuration.</p>"
    }
},
"Description": {
    "target": "com.amazonaws.licensemanager#String",
    "traits": {
        "smithy.api#documentation": "<p>New description of the license configuration.</p>"
    }
},
"ProductInformationList": {
    "target": "com.amazonaws.licensemanager#ProductInformationList",
    "traits": {
        "smithy.api#documentation": "<p>New product information.</p>"
    }
},
"DisassociateWhenNotFound": {
    "target": "com.amazonaws.licensemanager#BoxBoolean",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
                }
            }
        },
        "traits":
{
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanager#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanager#AuthorizationException"
            },
            {
                "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
            },
            {
                "target": "com.amazonaws.licensemanager#RateLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
            },
            {
                "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanager#ServerInternalException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanager#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates a report generator.</p>\n        <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
        }
    },
    "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
        "type": "structure",
        "members": {
            "LicenseManagerReportGeneratorArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
                    "smithy.api#required": {}
                }
            },
            "ReportGeneratorName": {
                "target": "com.amazonaws.licensemanager#ReportGeneratorName",
                "traits": {
                    "smithy.api#documentation": "<p>Name of the report generator.</p>",
                    "smithy.api#required": {}
                }
            },
            "Type": {
                "target": "com.amazonaws.licensemanager#ReportTypeList",
                "traits": {
                    "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n        <ul>\n        <li>\n
<p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n        </li>\n        <li>\n        <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n        </li>\n        </ul>",
                    "smithy.api#required": {}
                }
            },
            "ReportContext": {
                "target": "com.amazonaws.licensemanager#ReportContext",
                "traits": {
                    "smithy.api#documentation": "<p>The report context.</p>",
                    "smithy.api#required": {}
                }
            },
            "ReportFrequency": {
                "target": "com.amazonaws.licensemanager#ReportFrequency",
```

```
                    "traits":
{
                        "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
                        "smithy.api#required": {}
                    }
                },
                "ClientToken": {
                    "target": "com.amazonaws.licensemanager#ClientRequestToken",
                    "traits": {
                        "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
                        "smithy.api#required": {}
                    }
                },
                "Description": {
                    "target": "com.amazonaws.licensemanager#String",
                    "traits": {
                        "smithy.api#documentation": "<p>Description of the report generator.</p>"
                    }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanager#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanager#AuthorizationException"
                },
```

```
    {
      "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
    },

    {
      "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
    },
    {
      "target": "com.amazonaws.licensemanager#LicenseUsageException"
    },
    {
      "target": "com.amazonaws.licensemanager#RateLimitExceededException"
    },
    {
      "target": "com.amazonaws.licensemanager#ServerInternalException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified
Amazon Web Services resource.</p>\n      <p>You can update the license specifications of AMIs, instances, and
hosts.\n      You cannot update the license specifications for launch templates and CloudFormation templates, \n
 as they send license configurations to the operation that creates the resource.</p>"
  }
},

"com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanager#String",
      "traits": {
        "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services
resource.</p>",
        "smithy.api#required": {}
      }
    },
    "AddLicenseSpecifications": {
      "target": "com.amazonaws.licensemanager#LicenseSpecifications",
      "traits": {
        "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
      }
    },
    "RemoveLicenseSpecifications": {
      "target": "com.amazonaws.licensemanager#LicenseSpecifications",
      "traits": {
        "smithy.api#documentation":
"<p>ARNs of the license configurations to remove.</p>"
      }
```

```
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    }
  },
  "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanager#UpdateServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanager#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanager#AuthorizationException"
      },
      {
        "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
      },
      {
        "target": "com.amazonaws.licensemanager#RateLimitExceededException"
      },
      {
        "target": "com.amazonaws.licensemanager#ServerInternalException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
    }
  },
  "com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "S3BucketArn": {
        "target": "com.amazonaws.licensemanager#String",
        "traits": {
```

```
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
 Manager information is stored.</p>"
                }
            },
            "SnsTopicArn": {
                "target": "com.amazonaws.licensemanager#String",
                "traits": {
                    "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
                }
            },
            "OrganizationConfiguration": {
                "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
                "traits": {
                    "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
                }
            },
            "EnableCrossAccountsDiscovery": {
                "target": "com.amazonaws.licensemanager#BoxBoolean",
                "traits": {
                    "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"

    }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanager#UsageOperation": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        }
    }
},
"com.amazonaws.licensemanager#ValidationException": {
```

```
      "type": "structure",
      "members": {
        "Message": {
          "target": "com.amazonaws.licensemanager#Message"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

    "smithy.api#error": "client",
        "smithy.api#httpError": 400
      }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {
```

```go
	// Amazon Resource Name (ARN) of the license.
	//
	// This member is required.
	LicenseArn *string

	// Current version
	of the license.
	//
	// This member is required.
	SourceVersion *string

	noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

	// Date when the license is deleted.
	DeletionDate *string

	// License status.
	Status types.LicenseDeletionStatus

	// Metadata pertaining to the operation's result.
	ResultMetadata middleware.Metadata

	noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
	if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
		return err
	}
	err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
	if err != nil {
		return err
	}
	if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
		return fmt.Errorf("add protocol finalizers: %v", err)
	}

	if err = addlegacyEndpointContextSetter(stack,
	options); err != nil {
		return err
```

```
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddClientRequestIDMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddComputeContentLengthMiddleware(stack); err != nil {
    return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
    }
    if err = v4.AddComputePayloadSHA256Middleware(stack); err != nil {
    return err
    }
    if err = addRetryMiddlewares(stack, options); err != nil {
    return err
    }
    if err = awsmiddleware.AddRawResponseToMetadata(stack); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecordResponseTiming(stack); err != nil {
    return err
    }
    if err = addClientUserAgent(stack, options); err != nil {
    return err
    }
    if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
    }
    if
    err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
    }
    if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
    middleware.Before); err != nil {
    return err
    }
    if err = awsmiddleware.AddRecursionDetection(stack); err != nil {
    return err
    }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:        region,
ServiceID:     ServiceID,
OperationName: "DeleteLicense",
}
}
```

1.37 aws-sdk-go 1.48.9

1.37.1 Available under license :

```
{
 "version": "2.0",
 "service": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
 "operations": {
   "AssociateUser": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
   "DeregisterIdentityProvider": "<p>Deregisters the identity provider from providing user-based subscriptions.</p>",
   "DisassociateUser": "<p>Disassociates the user from an EC2 instance providing user-based subscriptions.</p>",
   "ListIdentityProviders":
"<p>Lists the identity providers for user-based subscriptions.</p>",
   "ListInstances": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
   "ListProductSubscriptions": "<p>Lists the user-based subscription products available from an identity provider.</p>",
   "ListUserAssociations": "<p>Lists user associations for an identity provider.</p>",
   "RegisterIdentityProvider": "<p>Registers an identity provider for user-based subscriptions.</p>",
   "StartProductSubscription": "<p>Starts a product subscription for a user with the specified identity provider.</p>
```

<note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as Pending billing status) in Amazon Web Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges in the <i>Amazon Web
 Services Billing User Guide</i>.</p> </note>",
   "StopProductSubscription": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
   "UpdateIdentityProviderSettings": "<p>Updates additional product configuration settings for the registered identity provider.</p>"
 },
 "shapes": {
  "AccessDeniedException": {
   "base": "<p>You don't have sufficient access to perform this action.</p>",
   "refs": {
   }
  },
  "ActiveDirectoryIdentityProvider": {
   "base": "<p>Details about an Active Directory identity provider.</p>",
   "refs": {
     "IdentityProvider$ActiveDirectoryIdentityProvider": "<p>An object that details an Active Directory identity provider.</p>"
   }
  },
  "AssociateUserRequest": {
   "base": null,
   "refs": {
   }
  },
  "AssociateUserResponse": {
   "base": null,
   "refs": {
   }
  },
  "BoxInteger": {
   "base": null,
   "refs": {
     "ListIdentityProvidersRequest$MaxResults":
 "<p>Maximum number of results to return in a single call.</p>",
     "ListInstancesRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
     "ListProductSubscriptionsRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
     "ListUserAssociationsRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>"
   }
  },
  "ConflictException": {
   "base": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
   "refs": {
   }

```
    },
  "DeregisterIdentityProviderRequest": {
    "base": null,
    "refs": {
    }
  },
  "DeregisterIdentityProviderResponse": {
    "base": null,
    "refs": {
    }
  },
  "DisassociateUserRequest": {
    "base": null,
    "refs": {
    }
  },
  "DisassociateUserResponse": {
    "base": null,
    "refs": {
    }
  },
  "Filter": {
    "base": "<p>A
```
 filter name and value pair that is used to return more specific results from a describe operation. Filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>",
```
    "refs": {
      "FilterList$member": null
    }
  },
  "FilterList": {
    "base": null,
    "refs": {
      "ListInstancesRequest$Filters": "<p>An array of structures that you can use to filter the results to those that
```
match one or more sets of key-value pairs that you specify.</p>",
```
      "ListProductSubscriptionsRequest$Filters": "<p>An array of structures that you can use to filter the results to
```
those that match one or more sets of key-value pairs that you specify.</p>",
```
      "ListUserAssociationsRequest$Filters": "<p>An array of structures that you can use to filter the results to those
```
that match one or more sets of key-value pairs that you specify.</p>"
```
    }
  },
  "IdentityProvider": {
    "base": "<p>Details about an identity provider.</p>",
    "refs": {
      "AssociateUserRequest$IdentityProvider": "<p>The identity provider of the user.</p>",
      "DeregisterIdentityProviderRequest$IdentityProvider": "<p>An object that specifies details for the identity
```
provider.</p>",
```
      "DisassociateUserRequest$IdentityProvider": "<p>An object that specifies details for the identity
```
provider.</p>",

    "IdentityProviderSummary$IdentityProvider": "<p>An object that specifies details for the identity provider.</p>",
     "InstanceUserSummary$IdentityProvider": "<p>An object that specifies details for the identity provider.</p>",
     "ListProductSubscriptionsRequest$IdentityProvider": "<p>An object that specifies details for the identity provider.</p>",
     "ListUserAssociationsRequest$IdentityProvider": "<p>An object that specifies details for the identity provider.</p>",
     "ProductUserSummary$IdentityProvider": "<p>An object that specifies details for the identity provider.</p>",
     "RegisterIdentityProviderRequest$IdentityProvider":
 "<p>An object that specifies details for the identity provider.</p>",
     "StartProductSubscriptionRequest$IdentityProvider": "<p>An object that specifies details for the identity provider.</p>",
     "StopProductSubscriptionRequest$IdentityProvider": "<p>An object that specifies details for the identity provider.</p>",
     "UpdateIdentityProviderSettingsRequest$IdentityProvider": null
   }
  },
  "IdentityProviderSummary": {
   "base": "<p>Describes an identity provider.</p>",
   "refs": {
     "DeregisterIdentityProviderResponse$IdentityProviderSummary": "<p>Metadata that describes the results of an identity provider operation.</p>",
     "IdentityProviderSummaryList$member": null,
     "RegisterIdentityProviderResponse$IdentityProviderSummary": "<p>Metadata that describes the results of an identity provider operation.</p>",
     "UpdateIdentityProviderSettingsResponse$IdentityProviderSummary": null

   }
  },
  "IdentityProviderSummaryList": {
   "base": null,
   "refs": {
     "ListIdentityProvidersResponse$IdentityProviderSummaries": "<p>Metadata that describes the list identity providers operation.</p>"
   }
  },
  "InstanceSummary": {
   "base": "<p>Describes an EC2 instance providing user-based subscriptions.</p>",
   "refs": {
    "InstanceSummaryList$member": null
   }
  },
  "InstanceSummaryList": {
   "base": null,
   "refs": {
    "ListInstancesResponse$InstanceSummaries": "<p>Metadata that describes the list instances operation.</p>"
   }
  },

"InstanceUserSummary": {
  "base": "<p>Describes users of an EC2 instance providing user-based subscriptions.</p>",
  "refs": {
    "AssociateUserResponse$InstanceUserSummary": "<p>Metadata that describes the associate user operation.</p>",
    "DisassociateUserResponse$InstanceUserSummary": "<p>Metadata that describes the associate user operation.</p>",

  "InstanceUserSummaryList$member": null
  }
},
"InstanceUserSummaryList": {
  "base": null,
  "refs": {
    "ListUserAssociationsResponse$InstanceUserSummaries": "<p>Metadata that describes the list user association operation.</p>"
  }
},
"InternalServerException": {
  "base": "<p>An exception occurred with the service.</p>",
  "refs": {
  }
},
"ListIdentityProvidersRequest": {
  "base": null,
  "refs": {
  }
},
"ListIdentityProvidersResponse": {
  "base": null,
  "refs": {
  }
},
"ListInstancesRequest": {
  "base": null,
  "refs": {
  }
},
"ListInstancesResponse": {
  "base": null,
  "refs": {
  }
},
"ListProductSubscriptionsRequest": {
  "base": null,
  "refs": {
  }
},

```
    "ListProductSubscriptionsResponse": {
     "base": null,
     "refs": {
     }
    },
    "ListUserAssociationsRequest": {
     "base": null,
     "refs": {
     }
    },
    "ListUserAssociationsResponse": {
     "base": null,
     "refs": {
     }
    },
    "ProductUserSummary": {
     "base": "<p>The summary of the user-based subscription products for a user.</p>",
     "refs": {
      "ProductUserSummaryList$member": null,
      "StartProductSubscriptionResponse$ProductUserSummary": "<p>Metadata that describes the start product
subscription operation.</p>",
      "StopProductSubscriptionResponse$ProductUserSummary": "<p>Metadata that describes the start product
subscription operation.</p>"
     }
    },
    "ProductUserSummaryList": {
     "base": null,
     "refs": {
      "ListProductSubscriptionsResponse$ProductUserSummaries": "<p>Metadata that describes the list product
subscriptions operation.</p>"
     }
    },
    "RegisterIdentityProviderRequest": {
     "base": null,
     "refs": {
     }
    },
    "RegisterIdentityProviderResponse": {
     "base": null,
     "refs": {
     }
    },
    "ResourceNotFoundException":
   {
     "base": "<p>The resource couldn't be found.</p>",
     "refs": {
     }
    },
```

"SecurityGroup": {
  "base": null,
  "refs": {
    "Settings$SecurityGroupId": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
    "UpdateSettings$SecurityGroupId": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
  }
},
"ServiceQuotaExceededException": {
  "base": "<p>The request failed because a service quota is exceeded.</p>",
  "refs": {
  }
},
"Settings": {
  "base": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication
 from resources in the VPC.</p>",
  "refs": {
    "IdentityProviderSummary$Settings": "<p>An object that details the registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>",
    "RegisterIdentityProviderRequest$Settings": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
  }
},
"SettingsSubnetsList": {
  "base": null,
  "refs": {
    "Settings$Subnets": "<p>The subnets defined for the registered identity provider.</p>"
  }
},
"StartProductSubscriptionRequest": {
  "base": null,
  "refs": {
  }
},
"StartProductSubscriptionResponse": {
  "base": null,
  "refs": {
  }
},
"StopProductSubscriptionRequest": {
  "base": null,
  "refs": {
  }
},
"StopProductSubscriptionResponse": {

      "base": null,
      "refs": {
      }
    },
    "String":
  {
      "base": null,
      "refs": {
        "AccessDeniedException$message": null,
        "ActiveDirectoryIdentityProvider$DirectoryId": "<p>The directory ID for an Active Directory identity provider.</p>",
        "AssociateUserRequest$Domain": "<p>The domain name of the user.</p>",
        "AssociateUserRequest$InstanceId": "<p>The ID of the EC2 instance, which provides user-based subscriptions.</p>",
        "AssociateUserRequest$Username": "<p>The user name from the identity provider for the user.</p>",
        "ConflictException$message": null,
        "DeregisterIdentityProviderRequest$Product": "<p>The name of the user-based subscription product.</p>",
        "DisassociateUserRequest$Domain": "<p>The domain name of the user.</p>",
        "DisassociateUserRequest$InstanceId": "<p>The ID of the EC2 instance, which provides user-based subscriptions.</p>",
        "DisassociateUserRequest$Username": "<p>The user name from the identity provider for the user.</p>",

      "Filter$Attribute": "<p>The name of an attribute to use as a filter.</p>",
        "Filter$Operation": "<p>The type of search (For example, eq, geq, leq)</p>",
        "Filter$Value": "<p>Value of the filter.</p>",
        "IdentityProviderSummary$FailureMessage": "<p>The failure message associated with an identity provider.</p>",
        "IdentityProviderSummary$Product": "<p>The name of the user-based subscription product.</p>",
        "IdentityProviderSummary$Status": "<p>The status of an identity provider.</p>",
        "InstanceSummary$InstanceId": "<p>The ID of the EC2 instance, which provides user-based subscriptions.</p>",
        "InstanceSummary$LastStatusCheckDate": "<p>The date of the last status check.</p>",
        "InstanceSummary$Status": "<p>The status of an EC2 instance resource.</p>",
        "InstanceSummary$StatusMessage": "<p>The status message for an EC2 instance.</p>",
        "InstanceUserSummary$AssociationDate": "<p>The date a user was associated with an
     EC2 instance.</p>",
        "InstanceUserSummary$DisassociationDate": "<p>The date a user was disassociated from an EC2 instance.</p>",
        "InstanceUserSummary$Domain": "<p>The domain name of the user.</p>",
        "InstanceUserSummary$InstanceId": "<p>The ID of the EC2 instance, which provides user-based subscriptions.</p>",
        "InstanceUserSummary$Status": "<p>The status of a user associated with an EC2 instance.</p>",
        "InstanceUserSummary$StatusMessage": "<p>The status message for users of an EC2 instance.</p>",
        "InstanceUserSummary$Username": "<p>The user name from the identity provider for the user.</p>",
        "InternalServerException$message": null,
        "ListIdentityProvidersRequest$NextToken": "<p>Token for the next set of results.</p>",
        "ListIdentityProvidersResponse$NextToken": "<p>Token for the next set of results.</p>",
        "ListInstancesRequest$NextToken": "<p>Token for the next set of results.</p>",

"ListInstancesResponse$NextToken":
"<p>Token for the next set of results.</p>",
        "ListProductSubscriptionsRequest$NextToken": "<p>Token for the next set of results.</p>",
        "ListProductSubscriptionsRequest$Product": "<p>The name of the user-based subscription product.</p>",
        "ListProductSubscriptionsResponse$NextToken": "<p>Token for the next set of results.</p>",
        "ListUserAssociationsRequest$InstanceId": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "ListUserAssociationsRequest$NextToken": "<p>Token for the next set of results.</p>",
        "ListUserAssociationsResponse$NextToken": "<p>Token for the next set of results.</p>",
        "ProductUserSummary$Domain": "<p>The domain name of the user.</p>",
        "ProductUserSummary$Product": "<p>The name of the user-based subscription product.</p>",
        "ProductUserSummary$Status": "<p>The status of a product for a user.</p>",
        "ProductUserSummary$StatusMessage": "<p>The status
message for a product for a user.</p>",
        "ProductUserSummary$SubscriptionEndDate": "<p>The end date of a subscription.</p>",
        "ProductUserSummary$SubscriptionStartDate": "<p>The start date of a subscription.</p>",
        "ProductUserSummary$Username": "<p>The user name from the identity provider of the user.</p>",
        "RegisterIdentityProviderRequest$Product": "<p>The name of the user-based subscription product.</p>",
        "ResourceNotFoundException$message": null,
        "ServiceQuotaExceededException$message": null,
        "StartProductSubscriptionRequest$Domain": "<p>The domain name of the user.</p>",
        "StartProductSubscriptionRequest$Product": "<p>The name of the user-based subscription product.</p>",
        "StartProductSubscriptionRequest$Username": "<p>The user name from the identity provider of the user.</p>",
        "StopProductSubscriptionRequest$Domain": "<p>The domain name of the user.</p>",
        "StopProductSubscriptionRequest$Product":
"<p>The name of the user-based subscription product.</p>",
        "StopProductSubscriptionRequest$Username": "<p>The user name from the identity provider for the
user.</p>",
        "StringList$member": null,
        "ThrottlingException$message": null,
        "UpdateIdentityProviderSettingsRequest$Product": "<p>The name of the user-based subscription
product.</p>",
        "ValidationException$message": null
      }
    },
    "StringList": {
      "base": null,
      "refs": {
        "InstanceSummary$Products": "<p>A list of provided user-based subscription products.</p>"
      }
    },
    "Subnet": {
      "base": null,
      "refs": {
        "SettingsSubnetsList$member": null,
        "Subnets$member": null
      }
    },

```
    "Subnets": {
      "base": null,
      "refs": {
        "UpdateSettings$AddSubnets": "<p>The ID of one or more subnets in which License Manager will create a
VPC endpoint for products that require connectivity to activation servers.</p>",
        "UpdateSettings$RemoveSubnets":
"<p>The ID of one or more subnets to remove.</p>"
      }
    },
    "ThrottlingException": {
      "base": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "refs": {
      }
    },
    "UpdateIdentityProviderSettingsRequest": {
      "base": null,
      "refs": {
      }
    },
    "UpdateIdentityProviderSettingsResponse": {
      "base": null,
      "refs": {
      }
    },
    "UpdateSettings": {
      "base": "<p>Updates the registered identity providers product related configuration settings such as the subnets
to provision VPC endpoints.</p>",
      "refs": {
        "UpdateIdentityProviderSettingsRequest$UpdateSettings": "<p>Updates the registered identity providers
product related configuration settings. You can update any combination of settings in a single operation such as
the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets
which you want to remove the VPC endpoints
 from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>"
      }
    },
    "ValidationException": {
      "base": "<p>A parameter is not valid.</p>",
      "refs": {
      }
    }
  }
}
{
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
```

```
          "documentation": "The AWS region used to dispatch the request.",
          "type": "String"
        },
        "UseDualStack": {
          "builtIn": "AWS::UseDualStack",
          "required": true,
          "default": false,
          "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
          "type": "Boolean"
        },
        "UseFIPS": {
          "builtIn": "AWS::UseFIPS",
          "required": true,
          "default": false,
          "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
          "type": "Boolean"
        },
        "Endpoint": {

          "builtIn": "SDK::Endpoint",
          "required": false,
          "documentation": "Override the endpoint used to send this request",
          "type": "String"
        }
      },
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "type": "tree",
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
```

```
                    "ref": "Endpoint"
                }
            ]
        },
        {
            "fn": "parseURL",

            "argv": [
                {
                    "ref": "Endpoint"
                }
            ],
            "assign": "url"
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
            "type": "error"
        },
        {
            "conditions": [],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],
```

```
                              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                              "type": "error"
                          },

                  {
                              "conditions": [],
                              "endpoint": {
                                "url": {
                                    "ref": "Endpoint"
                                },
                                "properties": {},
                                "headers": {}
                              },
                              "type": "endpoint"
                  }
              ]
          }
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                  {
                      "ref": "UseFIPS"
                  },
                  true
                ]

    },
            {
                "fn": "booleanEquals",
                "argv": [
                  {
                      "ref": "UseDualStack"
                  },
                  true
                ]
            }
        ],
        "type": "tree",
        "rules": [
            {
                "conditions": [
                  {
                      "fn": "booleanEquals",
                      "argv": [
```

```
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        }
                    ]
                },
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ]
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both",
            "type": "error"
```

```
                    }
                ]
            },

        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        }
                    ],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
```

```
                    ]
                },
                {

    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "type": "tree",
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,

{

                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "type": "tree",
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
```

```
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                              "properties": {},

                        "headers": {}
                            },
                          "type": "endpoint"
                        }
                      ]
                  },
                  {
                    "conditions": [],
                    "error": "DualStack is enabled but this partition does not support DualStack",
                    "type": "error"
                  }
                ]
            },
            {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ]
        }
    }
  ]
}
```

AWS SDK for Go

```
{
 "pagination": {
  "ListLinuxSubscriptionInstances": {
   "input_token": "NextToken",
   "output_token": "NextToken",
   "limit_key": "MaxResults",
   "result_key": "Instances"
  },
  "ListLinuxSubscriptions": {
   "input_token": "NextToken",
   "output_token": "NextToken",
   "limit_key": "MaxResults",
   "result_key": "Subscriptions"
  }
 }
```

```
        }
{
    "testCases": [
        {
            "documentation": "For region ap-south-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.ap-south-1.api.aws"
                }
            },
            "params": {
                "UseFIPS": true,
                "Region": "ap-south-1",
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region ap-south-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.ap-south-1.amazonaws.com"
                }
            },
            "params": {
                "UseFIPS": true,
                "Region": "ap-south-1",
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region ap-south-1 with FIPS disabled
and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.ap-south-1.api.aws"
                }
            },
            "params": {
                "UseFIPS": false,
                "Region": "ap-south-1",
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.ap-south-1.amazonaws.com"
```

```
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "ap-south-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-south-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.eu-south-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "eu-south-1",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region eu-south-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.eu-south-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "eu-south-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-south-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.eu-south-1.api.aws"
        }
      },

      "params": {
        "UseFIPS": false,
        "Region": "eu-south-1",
        "UseDualStack": true
      }
    },
```

```
{
    "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.eu-south-1.amazonaws.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "Region": "eu-south-1",
      "UseDualStack": false
    }
},
{
    "documentation": "For region ca-central-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.ca-central-1.api.aws"
      }
    },
    "params": {
      "UseFIPS": true,
      "Region": "ca-central-1",

      "UseDualStack": true
    }
},
{
    "documentation": "For region ca-central-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.ca-central-1.amazonaws.com"
      }
    },
    "params": {
      "UseFIPS": true,
      "Region": "ca-central-1",
      "UseDualStack": false
    }
},
{
    "documentation": "For region ca-central-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.ca-central-1.api.aws"
      }
    },
    "params": {
```

```
          "UseFIPS": false,
          "Region": "ca-central-1",
          "UseDualStack": true
        }
    },
    {
        "documentation":
"For region ca-central-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.ca-central-1.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": false,
          "Region": "ca-central-1",
          "UseDualStack": false
        }
    },
    {
        "documentation": "For region eu-central-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.eu-central-1.api.aws"
          }
        },
        "params": {
          "UseFIPS": true,
          "Region": "eu-central-1",
          "UseDualStack": true
        }
    },
    {
        "documentation": "For region eu-central-1 with FIPS enabled and DualStack disabled",
        "expect":
{
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.eu-central-1.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": true,
          "Region": "eu-central-1",
          "UseDualStack": false
        }
    },
    {
        "documentation": "For region eu-central-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.eu-central-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "eu-central-1",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.eu-central-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "eu-central-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-west-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-west-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "us-west-1",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-west-1.amazonaws.com"
        }
      },
      "params":
{
        "UseFIPS": true,
        "Region": "us-west-1",
```

```
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-west-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-west-1.api.aws"
                }
            },
            "params": {
                "UseFIPS": false,
                "Region": "us-west-1",
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-west-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-west-1.amazonaws.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "Region": "us-west-1",
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-west-2 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-west-2.api.aws"
                }
            },
            "params": {
                "UseFIPS": true,
                "Region": "us-west-2",
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-west-2.amazonaws.com"
                }
```

```
      },
      "params": {
        "UseFIPS": true,
        "Region": "us-west-2",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-west-2 with FIPS disabled
and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-west-2.api.aws"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "us-west-2",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-west-2.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "us-west-2",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region af-south-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.af-south-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "af-south-1",
        "UseDualStack": true
      }
    },
    {
```

```
      "documentation": "For region af-south-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.af-south-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "af-south-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region af-south-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.af-south-1.api.aws"
        }
      },

"params": {
        "UseFIPS": false,
        "Region": "af-south-1",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.af-south-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "af-south-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.eu-north-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
```

```
                    "Region": "eu-north-1",

  "UseDualStack": true
        }
    },
    {
        "documentation": "For region eu-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.eu-north-1.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": true,
          "Region": "eu-north-1",
          "UseDualStack": false
        }
    },
    {
        "documentation": "For region eu-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.eu-north-1.api.aws"
          }
        },
        "params": {
          "UseFIPS": false,
          "Region": "eu-north-1",
          "UseDualStack": true
        }
    },
    {
        "documentation": "For region eu-north-1
with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.eu-north-1.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": false,
          "Region": "eu-north-1",
          "UseDualStack": false
        }
    },
    {
        "documentation": "For region eu-west-3 with FIPS enabled and DualStack enabled",
        "expect": {
```

```
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.eu-west-3.api.aws"
          }
        },
        "params": {
          "UseFIPS": true,
          "Region": "eu-west-3",
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region eu-west-3 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {


            "url": "https://license-manager-user-subscriptions-fips.eu-west-3.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": true,
          "Region": "eu-west-3",
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-west-3 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.eu-west-3.api.aws"
          }
        },
        "params": {
          "UseFIPS": false,
          "Region": "eu-west-3",
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.eu-west-3.amazonaws.com"


}
      },
      "params": {
        "UseFIPS": false,
        "Region": "eu-west-3",
```

```
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-west-2 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.eu-west-2.api.aws"
          }
        },
        "params": {
          "UseFIPS": true,
          "Region": "eu-west-2",
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region eu-west-2 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.eu-west-2.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": true,
          "Region": "eu-west-2",
           "UseDualStack": false
        }
      },
      {
        "documentation": "For region eu-west-2 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.eu-west-2.api.aws"
          }
        },
        "params": {
          "UseFIPS": false,
          "Region": "eu-west-2",
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.eu-west-2.amazonaws.com"
          }
```

```
      },
      "params": {
        "UseFIPS": false,
        "Region": "eu-west-2",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For
region eu-west-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.eu-west-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "eu-west-1",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region eu-west-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.eu-west-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "eu-west-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region eu-west-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions.eu-west-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "eu-west-1",
        "UseDualStack": true
      }
    },
```

```
{
    "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.eu-west-1.amazonaws.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "Region": "eu-west-1",
      "UseDualStack": false
    }
},
{
    "documentation": "For region ap-northeast-3 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.ap-northeast-3.api.aws"
      }
    },
    "params": {
      "UseFIPS": true,
      "Region": "ap-northeast-3",
      "UseDualStack": true
    }
},
{
    "documentation": "For region ap-northeast-3 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.ap-northeast-3.amazonaws.com"
      }
    },
    "params": {
      "UseFIPS": true,
      "Region": "ap-northeast-3",
      "UseDualStack": false
    }
},
{
    "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.ap-northeast-3.api.aws"
      }
    },
    "params": {
      "UseFIPS":
```

```
false,
        "Region": "ap-northeast-3",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.ap-northeast-3.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "ap-northeast-3",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-2 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.ap-northeast-2.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "ap-northeast-2",
        "UseDualStack":
true
      }
    },
    {
      "documentation": "For region ap-northeast-2 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.ap-northeast-2.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "ap-northeast-2",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack enabled",
      "expect": {
```

```
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.ap-northeast-2.api.aws"
      }
    },
    "params": {
      "UseFIPS": false,
      "Region": "ap-northeast-2",
      "UseDualStack": true
    }
  },
  {
    "documentation":
"For region ap-northeast-2 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.ap-northeast-2.amazonaws.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "Region": "ap-northeast-2",
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region ap-northeast-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.ap-northeast-1.api.aws"
      }
    },
    "params": {
      "UseFIPS": true,
      "Region": "ap-northeast-1",
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region ap-northeast-1 with FIPS enabled and DualStack disabled",

    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.ap-northeast-1.amazonaws.com"
      }
    },
    "params": {
      "UseFIPS": true,
      "Region": "ap-northeast-1",
```

```
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.ap-northeast-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "ap-northeast-1",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.ap-northeast-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "ap-northeast-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region me-south-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.me-south-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "me-south-1",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region me-south-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.me-south-1.amazonaws.com"
        }
```

```
        },
        "params": {
          "UseFIPS": true,
          "Region": "me-south-1",
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region me-south-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.me-south-1.api.aws"
          }
        },
        "params": {
          "UseFIPS": false,
          "Region": "me-south-1",
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.me-south-1.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": false,
          "Region": "me-south-1",
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region sa-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.sa-east-1.api.aws"
          }
        },
        "params": {
          "UseFIPS": true,
          "Region": "sa-east-1",
          "UseDualStack": true
        }
      },
      {
```

```
      "documentation": "For region sa-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.sa-east-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "sa-east-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For
region sa-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.sa-east-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "sa-east-1",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.sa-east-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "sa-east-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.ap-east-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
```

```
          "Region": "ap-east-1",
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region ap-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.ap-east-1.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": true,
          "Region": "ap-east-1",
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region ap-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.ap-east-1.api.aws"

}
      },
      "params": {
        "UseFIPS": false,
        "Region": "ap-east-1",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.ap-east-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "ap-east-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-southeast-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
```

```
          "url": "https://license-manager-user-subscriptions-fips.ap-southeast-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region":
"ap-southeast-1",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region ap-southeast-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.ap-southeast-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "ap-southeast-1",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.ap-southeast-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "ap-southeast-1",
        "UseDualStack": true
      }

    },
    {
      "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.ap-southeast-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "ap-southeast-1",
        "UseDualStack": false
```

```
      }
    },
    {
      "documentation": "For region ap-southeast-2 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.ap-southeast-2.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "ap-southeast-2",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region ap-southeast-2 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.ap-southeast-2.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": true,
        "Region": "ap-southeast-2",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.ap-southeast-2.api.aws"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "ap-southeast-2",
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions.ap-southeast-2.amazonaws.com"
```

```
      }
    },
    "params": {
      "UseFIPS": false,
      "Region": "ap-southeast-2",
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
      }
    },
    "params": {
      "UseFIPS": true,
      "Region": "us-east-1",
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "UseFIPS": true,
      "Region": "us-east-1",
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {
      "UseFIPS": false,
      "Region": "us-east-1",
      "UseDualStack": true
    }
  },
  {
```

         "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
         "expect": {
            "endpoint": {
               "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
         },
         "params": {
            "UseFIPS": false,

"Region": "us-east-1",
            "UseDualStack": false
         }
      },
      {
         "documentation": "For region us-east-2 with FIPS enabled and DualStack enabled",
         "expect": {
            "endpoint": {
               "url": "https://license-manager-user-subscriptions-fips.us-east-2.api.aws"
            }
         },
         "params": {
            "UseFIPS": true,
            "Region": "us-east-2",
            "UseDualStack": true
         }
      },
      {
         "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
         "expect": {
            "endpoint": {
               "url": "https://license-manager-user-subscriptions-fips.us-east-2.amazonaws.com"
            }
         },
         "params": {
            "UseFIPS": true,
            "Region": "us-east-2",
            "UseDualStack": false
         }
      },
      {

   "documentation": "For region us-east-2 with FIPS disabled and DualStack enabled",
         "expect": {
            "endpoint": {
               "url": "https://license-manager-user-subscriptions.us-east-2.api.aws"
            }
         },
         "params": {

```
          "UseFIPS": false,
          "Region": "us-east-2",
          "UseDualStack": true
        }
    },
    {
      "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-2.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "us-east-2",
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack disabled",
      "expect":

{
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "Region": "us-east-1",
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {
        "UseFIPS": true,
        "Region": "us-east-1",
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
```

```
          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"

        },
          "params": {
            "UseFIPS": false,
            "Region": "us-east-1",
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
      }
  ],
  "version": "1.0"
}
{
 "version":"2.0",
 "metadata":{
  "apiVersion":"2018-08-01",
  "endpointPrefix":"license-manager",
  "jsonVersion":"1.1",
  "protocol":"json",
  "serviceFullName":"AWS License Manager",
  "serviceId":"License Manager",
  "signatureVersion":"v4",
  "targetPrefix":"AWSLicenseManager",
  "uid":"license-manager-2018-08-01"
 },
 "operations":{
  "AcceptGrant":{
   "name":"AcceptGrant",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"AcceptGrantRequest"},
   "output":{"shape":"AcceptGrantResponse"},
   "errors":[
    {"shape":"ValidationException"},
    {"shape":"InvalidParameterValueException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"},
    {"shape":"ResourceLimitExceededException"},
    {"shape":"ServerInternalException"}
   ]
  },
  "CheckInLicense":{
   "name":"CheckInLicense",
   "http":{
```

```
      "method":"POST",
      "requestUri":"/"
     },
     "input":{"shape":"CheckInLicenseRequest"},
     "output":{"shape":"CheckInLicenseResponse"},
     "errors":[
      {"shape":"ValidationException"},
      {"shape":"InvalidParameterValueException"},
      {"shape":"ResourceNotFoundException"},
      {"shape":"ConflictException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"},
      {"shape":"ServerInternalException"}
     ]
    },
    "CheckoutBorrowLicense":{
     "name":"CheckoutBorrowLicense",
     "http":{
      "method":"POST",
      "requestUri":"/"
     },
     "input":{"shape":"CheckoutBorrowLicenseRequest"},
     "output":{"shape":"CheckoutBorrowLicenseResponse"},
     "errors":[
      {"shape":"ValidationException"},
      {"shape":"InvalidParameterValueException"},
      {"shape":"ResourceNotFoundException"},
      {"shape":"NoEntitlementsAllowedException"},
       {"shape":"EntitlementNotAllowedException"},
      {"shape":"UnsupportedDigitalSignatureMethodException"},
      {"shape":"RedirectException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"},
      {"shape":"ServerInternalException"}
     ]
    },
    "CheckoutLicense":{
     "name":"CheckoutLicense",
     "http":{
      "method":"POST",
      "requestUri":"/"
     },
     "input":{"shape":"CheckoutLicenseRequest"},
     "output":{"shape":"CheckoutLicenseResponse"},
     "errors":[
      {"shape":"ValidationException"},
```

```
      {"shape":"InvalidParameterValueException"},
      {"shape":"ResourceNotFoundException"},
      {"shape":"NoEntitlementsAllowedException"},
      {"shape":"UnsupportedDigitalSignatureMethodException"},
      {"shape":"RedirectException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
       {"shape":"RateLimitExceededException"},
      {"shape":"ServerInternalException"}
    ]
  },
  "CreateGrant":{
   "name":"CreateGrant",
   "http":{
     "method":"POST",
     "requestUri":"/"
   },
   "input":{"shape":"CreateGrantRequest"},
   "output":{"shape":"CreateGrantResponse"},
   "errors":[
     {"shape":"ValidationException"},
     {"shape":"InvalidParameterValueException"},
     {"shape":"ResourceLimitExceededException"},
     {"shape":"ServerInternalException"},
     {"shape":"AuthorizationException"},
     {"shape":"RateLimitExceededException"},
     {"shape":"AccessDeniedException"}
   ]
  },
  "CreateGrantVersion":{
   "name":"CreateGrantVersion",
   "http":{
     "method":"POST",
     "requestUri":"/"
   },
   "input":{"shape":"CreateGrantVersionRequest"},
   "output":{"shape":"CreateGrantVersionResponse"},
   "errors":[
     {"shape":"ValidationException"},
     {"shape":"InvalidParameterValueException"},
      {"shape":"ResourceLimitExceededException"},
     {"shape":"ServerInternalException"},
     {"shape":"AuthorizationException"},
     {"shape":"AccessDeniedException"},
     {"shape":"RateLimitExceededException"}
   ]
  },
  "CreateLicense":{
```

```
   "name":"CreateLicense",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"CreateLicenseRequest"},
   "output":{"shape":"CreateLicenseResponse"},
   "errors":[
    {"shape":"ValidationException"},
    {"shape":"InvalidParameterValueException"},
    {"shape":"RedirectException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"},
    {"shape":"ServerInternalException"}
   ]
  },
  "CreateLicenseConfiguration":{
   "name":"CreateLicenseConfiguration",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },

   "input":{"shape":"CreateLicenseConfigurationRequest"},
   "output":{"shape":"CreateLicenseConfigurationResponse"},
   "errors":[
    {"shape":"InvalidParameterValueException"},
    {"shape":"ServerInternalException"},
    {"shape":"ResourceLimitExceededException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"}
   ]
  },
  "CreateLicenseConversionTaskForResource":{
   "name":"CreateLicenseConversionTaskForResource",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"CreateLicenseConversionTaskForResourceRequest"},
   "output":{"shape":"CreateLicenseConversionTaskForResourceResponse"},
   "errors":[
    {"shape":"InvalidParameterValueException"},
    {"shape":"ValidationException"},
    {"shape":"ServerInternalException"},
    {"shape":"AuthorizationException"},
```

```
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
     ]
    },
    "CreateLicenseManagerReportGenerator":{
      "name":"CreateLicenseManagerReportGenerator",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"CreateLicenseManagerReportGeneratorRequest"},
      "output":{"shape":"CreateLicenseManagerReportGeneratorResponse"},
      "errors":[
        {"shape":"ValidationException"},
        {"shape":"InvalidParameterValueException"},
        {"shape":"ResourceLimitExceededException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"RateLimitExceededException"},
        {"shape":"AccessDeniedException"},
        {"shape":"ResourceNotFoundException"}
     ]
    },
    "CreateLicenseVersion":{
      "name":"CreateLicenseVersion",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"CreateLicenseVersionRequest"},
      "output":{"shape":"CreateLicenseVersionResponse"},
       "errors":[
        {"shape":"ValidationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"ResourceNotFoundException"},
        {"shape":"RedirectException"},
        {"shape":"ConflictException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"RateLimitExceededException"}
     ]
    },
    "CreateToken":{
      "name":"CreateToken",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
```

```
    "input":{"shape":"CreateTokenRequest"},
    "output":{"shape":"CreateTokenResponse"},
    "errors":[
      {"shape":"ValidationException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"},
      {"shape":"ResourceLimitExceededException"},
      {"shape":"ServerInternalException"},
      {"shape":"ResourceNotFoundException"},
      {"shape":"RedirectException"}
    ]
  },
  "DeleteGrant":{
    "name":"DeleteGrant",
    "http":{
      "method":"POST",
      "requestUri":"/"
    },
    "input":{"shape":"DeleteGrantRequest"},
    "output":{"shape":"DeleteGrantResponse"},
    "errors":[
      {"shape":"ValidationException"},
      {"shape":"InvalidParameterValueException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"},
      {"shape":"ResourceLimitExceededException"},
      {"shape":"ServerInternalException"}
    ]
  },
  "DeleteLicense":{
    "name":"DeleteLicense",
    "http":{
      "method":"POST",
      "requestUri":"/"
    },
    "input":{"shape":"DeleteLicenseRequest"},
    "output":{"shape":"DeleteLicenseResponse"},
    "errors":[
      {"shape":"ValidationException"},
      {"shape":"InvalidParameterValueException"},
      {"shape":"RedirectException"},
      {"shape":"ConflictException"},

      {"shape":"ServerInternalException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
```

```
        {"shape":"RateLimitExceededException"}
      ]
    },
    "DeleteLicenseConfiguration":{
      "name":"DeleteLicenseConfiguration",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"DeleteLicenseConfigurationRequest"},
      "output":{"shape":"DeleteLicenseConfigurationResponse"},
      "errors":[
        {"shape":"InvalidParameterValueException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    },
    "DeleteLicenseManagerReportGenerator":{
      "name":"DeleteLicenseManagerReportGenerator",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"DeleteLicenseManagerReportGeneratorRequest"},
      "output":{"shape":"DeleteLicenseManagerReportGeneratorResponse"},
      "errors":[
        {"shape":"ValidationException"},
        {"shape":"InvalidParameterValueException"},
        {"shape":"ResourceLimitExceededException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"RateLimitExceededException"},
        {"shape":"AccessDeniedException"},
        {"shape":"ResourceNotFoundException"}
      ]
    },
    "DeleteToken":{
      "name":"DeleteToken",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"DeleteTokenRequest"},
      "output":{"shape":"DeleteTokenResponse"},
      "errors":[
        {"shape":"ValidationException"},
```

```
          {"shape":"AuthorizationException"},
          {"shape":"AccessDeniedException"},
          {"shape":"RateLimitExceededException"},
          {"shape":"ServerInternalException"},
          {"shape":"ResourceNotFoundException"},
          {"shape":"RedirectException"}
        ]
      },
      "ExtendLicenseConsumption":{
        "name":"ExtendLicenseConsumption",
        "http":{
          "method":"POST",
          "requestUri":"/"
        },
        "input":{"shape":"ExtendLicenseConsumptionRequest"},
        "output":{"shape":"ExtendLicenseConsumptionResponse"},
        "errors":[
          {"shape":"ValidationException"},
          {"shape":"InvalidParameterValueException"},
          {"shape":"AuthorizationException"},
          {"shape":"AccessDeniedException"},
          {"shape":"RateLimitExceededException"},
          {"shape":"ServerInternalException"},
          {"shape":"ResourceNotFoundException"}
        ]
      },
      "GetAccessToken":{
        "name":"GetAccessToken",
        "http":{
          "method":"POST",
          "requestUri":"/"
        },
        "input":{"shape":"GetAccessTokenRequest"},
        "output":{"shape":"GetAccessTokenResponse"},
        "errors":[
          {"shape":"ValidationException"},
          {"shape":"AuthorizationException"},
          {"shape":"AccessDeniedException"},

{"shape":"RateLimitExceededException"},
          {"shape":"ServerInternalException"}
        ]
      },
      "GetGrant":{
        "name":"GetGrant",
        "http":{
          "method":"POST",
          "requestUri":"/"
```

```
      },
      "input":{"shape":"GetGrantRequest"},
      "output":{"shape":"GetGrantResponse"},
      "errors":[
        {"shape":"ValidationException"},
        {"shape":"InvalidParameterValueException"},
        {"shape":"ResourceLimitExceededException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    },
    "GetLicense":{
      "name":"GetLicense",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"GetLicenseRequest"},
      "output":{"shape":"GetLicenseResponse"},
      "errors":[
        {"shape":"ValidationException"},
        {"shape":"InvalidParameterValueException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"},
        {"shape":"ServerInternalException"}
      ]
    },
    "GetLicenseConfiguration":{
      "name":"GetLicenseConfiguration",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"GetLicenseConfigurationRequest"},
      "output":{"shape":"GetLicenseConfigurationResponse"},
      "errors":[
        {"shape":"InvalidParameterValueException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    },
    "GetLicenseConversionTask":{
      "name":"GetLicenseConversionTask",
```

```
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"GetLicenseConversionTaskRequest"},
      "output":{"shape":"GetLicenseConversionTaskResponse"},
      "errors":[
        {"shape":"InvalidParameterValueException"},
         {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    },
    "GetLicenseManagerReportGenerator":{
      "name":"GetLicenseManagerReportGenerator",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"GetLicenseManagerReportGeneratorRequest"},
      "output":{"shape":"GetLicenseManagerReportGeneratorResponse"},
      "errors":[
        {"shape":"ValidationException"},
        {"shape":"InvalidParameterValueException"},
        {"shape":"ResourceLimitExceededException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"RateLimitExceededException"},
        {"shape":"AccessDeniedException"},
        {"shape":"ResourceNotFoundException"}
      ]
    },
    "GetLicenseUsage":{
      "name":"GetLicenseUsage",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"GetLicenseUsageRequest"},
      "output":{"shape":"GetLicenseUsageResponse"},
      "errors":[
        {"shape":"ValidationException"},
        {"shape":"InvalidParameterValueException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"},
        {"shape":"ServerInternalException"}
```

```
      ]
    },
    "GetServiceSettings":{
      "name":"GetServiceSettings",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"GetServiceSettingsRequest"},
      "output":{"shape":"GetServiceSettingsResponse"},
      "errors":[
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    },
    "ListAssociationsForLicenseConfiguration":{
      "name":"ListAssociationsForLicenseConfiguration",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"ListAssociationsForLicenseConfigurationRequest"},
      "output":{"shape":"ListAssociationsForLicenseConfigurationResponse"},
      "errors":[
        {"shape":"InvalidParameterValueException"},
        {"shape":"FilterLimitExceededException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    },
    "ListDistributedGrants":{
      "name":"ListDistributedGrants",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"ListDistributedGrantsRequest"},
      "output":{"shape":"ListDistributedGrantsResponse"},
      "errors":[
        {"shape":"ValidationException"},
        {"shape":"InvalidParameterValueException"},
        {"shape":"ResourceLimitExceededException"},
        {"shape":"ServerInternalException"},
```

```
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"}
   ]
  },
  "ListFailuresForLicenseConfigurationOperations":{
   "name":"ListFailuresForLicenseConfigurationOperations",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"ListFailuresForLicenseConfigurationOperationsRequest"},
   "output":{"shape":"ListFailuresForLicenseConfigurationOperationsResponse"},
   "errors":[
    {"shape":"InvalidParameterValueException"},
    {"shape":"ServerInternalException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"}
   ]
  },
  "ListLicenseConfigurations":{
   "name":"ListLicenseConfigurations",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"ListLicenseConfigurationsRequest"},
   "output":{"shape":"ListLicenseConfigurationsResponse"},
   "errors":[
    {"shape":"InvalidParameterValueException"},
    {"shape":"ServerInternalException"},
    {"shape":"FilterLimitExceededException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"}
   ]
  },
  "ListLicenseConversionTasks":{
   "name":"ListLicenseConversionTasks",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"ListLicenseConversionTasksRequest"},
   "output":{"shape":"ListLicenseConversionTasksResponse"},
   "errors":[
    {"shape":"InvalidParameterValueException"},
```

```
    {"shape":"ServerInternalException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"}
   ]
  },
  "ListLicenseManagerReportGenerators":{
   "name":"ListLicenseManagerReportGenerators",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"ListLicenseManagerReportGeneratorsRequest"},
   "output":{"shape":"ListLicenseManagerReportGeneratorsResponse"},
   "errors":[
    {"shape":"ValidationException"},
    {"shape":"InvalidParameterValueException"},
    {"shape":"ResourceLimitExceededException"},
    {"shape":"ServerInternalException"},
    {"shape":"AuthorizationException"},
    {"shape":"RateLimitExceededException"},
    {"shape":"AccessDeniedException"},
    {"shape":"ResourceNotFoundException"}
   ]
  },
  "ListLicenseSpecificationsForResource":{
   "name":"ListLicenseSpecificationsForResource",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"ListLicenseSpecificationsForResourceRequest"},
   "output":{"shape":"ListLicenseSpecificationsForResourceResponse"},
   "errors":[
    {"shape":"InvalidParameterValueException"},
     {"shape":"ServerInternalException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"}
   ]
  },
  "ListLicenseVersions":{
   "name":"ListLicenseVersions",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"ListLicenseVersionsRequest"},
```

```
   "output":{"shape":"ListLicenseVersionsResponse"},
   "errors":[
    {"shape":"InvalidParameterValueException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"},
    {"shape":"ServerInternalException"}
   ]
  },
  "ListLicenses":{
   "name":"ListLicenses",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"ListLicensesRequest"},
   "output":{"shape":"ListLicensesResponse"},
   "errors":[
    {"shape":"ValidationException"},

    {"shape":"InvalidParameterValueException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"},
    {"shape":"ServerInternalException"}
   ]
  },
  "ListReceivedGrants":{
   "name":"ListReceivedGrants",
   "http":{
    "method":"POST",
    "requestUri":"/"
   },
   "input":{"shape":"ListReceivedGrantsRequest"},
   "output":{"shape":"ListReceivedGrantsResponse"},
   "errors":[
    {"shape":"ValidationException"},
    {"shape":"InvalidParameterValueException"},
    {"shape":"ResourceLimitExceededException"},
    {"shape":"ServerInternalException"},
    {"shape":"AuthorizationException"},
    {"shape":"AccessDeniedException"},
    {"shape":"RateLimitExceededException"}
   ]
  },
  "ListReceivedGrantsForOrganization":{
   "name":"ListReceivedGrantsForOrganization",
   "http":{
```

```
      "method":"POST",
      "requestUri":"/"
     },
     "input":{"shape":"ListReceivedGrantsForOrganizationRequest"},
     "output":{"shape":"ListReceivedGrantsForOrganizationResponse"},
     "errors":[
      {"shape":"ValidationException"},
      {"shape":"InvalidParameterValueException"},
      {"shape":"ResourceLimitExceededException"},
      {"shape":"ServerInternalException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"}
     ]
    },
    "ListReceivedLicenses":{
     "name":"ListReceivedLicenses",
     "http":{
      "method":"POST",
      "requestUri":"/"
     },
     "input":{"shape":"ListReceivedLicensesRequest"},
     "output":{"shape":"ListReceivedLicensesResponse"},
     "errors":[
      {"shape":"ValidationException"},
      {"shape":"InvalidParameterValueException"},
      {"shape":"ResourceLimitExceededException"},
      {"shape":"ServerInternalException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"}
     ]
    },
    "ListReceivedLicensesForOrganization":{
     "name":"ListReceivedLicensesForOrganization",
     "http":{
      "method":"POST",
      "requestUri":"/"
     },
     "input":{"shape":"ListReceivedLicensesForOrganizationRequest"},
     "output":{"shape":"ListReceivedLicensesForOrganizationResponse"},
     "errors":[
      {"shape":"ValidationException"},
      {"shape":"InvalidParameterValueException"},
      {"shape":"ResourceLimitExceededException"},
      {"shape":"ServerInternalException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
```

```
      {"shape":"RateLimitExceededException"}
    ]
   },
   "ListResourceInventory":{
    "name":"ListResourceInventory",
    "http":{
      "method":"POST",
      "requestUri":"/"
    },
    "input":{"shape":"ListResourceInventoryRequest"},
    "output":{"shape":"ListResourceInventoryResponse"},
    "errors":[
      {"shape":"InvalidParameterValueException"},
      {"shape":"ServerInternalException"},
      {"shape":"FilterLimitExceededException"},
      {"shape":"FailedDependencyException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"}
    ]
   },
   "ListTagsForResource":{
    "name":"ListTagsForResource",
    "http":{
      "method":"POST",
      "requestUri":"/"
    },
    "input":{"shape":"ListTagsForResourceRequest"},
    "output":{"shape":"ListTagsForResourceResponse"},
    "errors":[
      {"shape":"InvalidParameterValueException"},
      {"shape":"ServerInternalException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"}
    ]
   },
   "ListTokens":{
    "name":"ListTokens",
    "http":{
      "method":"POST",

"requestUri":"/"
    },
    "input":{"shape":"ListTokensRequest"},
    "output":{"shape":"ListTokensResponse"},
    "errors":[
      {"shape":"ValidationException"},
```

```
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"},
      {"shape":"ServerInternalException"}
     ]
    },
    "ListUsageForLicenseConfiguration":{
     "name":"ListUsageForLicenseConfiguration",
     "http":{
      "method":"POST",
      "requestUri":"/"
     },
     "input":{"shape":"ListUsageForLicenseConfigurationRequest"},
     "output":{"shape":"ListUsageForLicenseConfigurationResponse"},
     "errors":[
      {"shape":"InvalidParameterValueException"},
      {"shape":"FilterLimitExceededException"},
      {"shape":"ServerInternalException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"}
     ]
    },
    "RejectGrant":{
     "name":"RejectGrant",
     "http":{
      "method":"POST",
      "requestUri":"/"
     },
     "input":{"shape":"RejectGrantRequest"},
     "output":{"shape":"RejectGrantResponse"},
     "errors":[
      {"shape":"ValidationException"},
      {"shape":"InvalidParameterValueException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"},
      {"shape":"ResourceLimitExceededException"},
      {"shape":"ServerInternalException"}
     ]
    },
    "TagResource":{
     "name":"TagResource",
     "http":{
      "method":"POST",
      "requestUri":"/"
     },
     "input":{"shape":"TagResourceRequest"},
```

```
    "output":{"shape":"TagResourceResponse"},
    "errors":[
      {"shape":"InvalidParameterValueException"},
      {"shape":"ServerInternalException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"}
    ]
  },
  "UntagResource":{
    "name":"UntagResource",
    "http":{
      "method":"POST",
      "requestUri":"/"
    },
    "input":{"shape":"UntagResourceRequest"},
    "output":{"shape":"UntagResourceResponse"},
    "errors":[
      {"shape":"InvalidParameterValueException"},
      {"shape":"ServerInternalException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"}
    ]
  },
  "UpdateLicenseConfiguration":{
    "name":"UpdateLicenseConfiguration",
    "http":{
      "method":"POST",
      "requestUri":"/"
    },
    "input":{"shape":"UpdateLicenseConfigurationRequest"},
    "output":{"shape":"UpdateLicenseConfigurationResponse"},
    "errors":[
      {"shape":"InvalidParameterValueException"},
      {"shape":"ServerInternalException"},
      {"shape":"AuthorizationException"},
      {"shape":"AccessDeniedException"},
      {"shape":"RateLimitExceededException"},
      {"shape":"ResourceLimitExceededException"}
    ]
  },
  "UpdateLicenseManagerReportGenerator":{
    "name":"UpdateLicenseManagerReportGenerator",
    "http":{
      "method":"POST",
      "requestUri":"/"
    },
```

```json
      "input":{"shape":"UpdateLicenseManagerReportGeneratorRequest"},
      "output":{"shape":"UpdateLicenseManagerReportGeneratorResponse"},
      "errors":[
        {"shape":"ValidationException"},
        {"shape":"InvalidParameterValueException"},
        {"shape":"RateLimitExceededException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"ResourceNotFoundException"},
        {"shape":"ResourceLimitExceededException"}
      ]
    },
    "UpdateLicenseSpecificationsForResource":{
      "name":"UpdateLicenseSpecificationsForResource",
      "http":{
        "method":"POST",
        "requestUri":"/"

    },
      "input":{"shape":"UpdateLicenseSpecificationsForResourceRequest"},
      "output":{"shape":"UpdateLicenseSpecificationsForResourceResponse"},
      "errors":[
        {"shape":"InvalidParameterValueException"},
        {"shape":"InvalidResourceStateException"},
        {"shape":"LicenseUsageException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    },
    "UpdateServiceSettings":{
      "name":"UpdateServiceSettings",
      "http":{
        "method":"POST",
        "requestUri":"/"
      },
      "input":{"shape":"UpdateServiceSettingsRequest"},
      "output":{"shape":"UpdateServiceSettingsResponse"},
      "errors":[
        {"shape":"InvalidParameterValueException"},
        {"shape":"ServerInternalException"},
        {"shape":"AuthorizationException"},
        {"shape":"AccessDeniedException"},
        {"shape":"RateLimitExceededException"}
      ]
    }
```

```
    },
    "shapes":{
     "AcceptGrantRequest":{
       "type":"structure",
       "required":["GrantArn"],
       "members":{
         "GrantArn":{"shape":"Arn"}
       }
     },
     "AcceptGrantResponse":{
       "type":"structure",
       "members":{
         "GrantArn":{"shape":"Arn"},
         "Status":{"shape":"GrantStatus"},
         "Version":{"shape":"String"}
       }
     },
     "AccessDeniedException":{
       "type":"structure",
       "members":{
         "Message":{"shape":"Message"}
       },
       "exception":true
     },
     "ActivationOverrideBehavior":{
       "type":"string",
       "enum":[
         "DISTRIBUTED_GRANTS_ONLY",
         "ALL_GRANTS_PERMITTED_BY_ISSUER"
       ]
     },
     "AllowedOperation":{
       "type":"string",
       "enum":[
         "CreateGrant",
         "CheckoutLicense",
         "CheckoutBorrowLicense",
         "CheckInLicense",
         "ExtendConsumptionLicense",
         "ListPurchasedLicenses",
         "CreateToken"
       ]
     },
     "AllowedOperationList":{
       "type":"list",
       "member":{"shape":"AllowedOperation"},
       "max":7,
       "min":1
```

```
    },
    "Arn":{
      "type":"string",
      "max":2048,
      "pattern":"^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9_/.-]{0,63}:[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,1023}$"
    },
    "ArnList":{
      "type":"list",
      "member":{"shape":"Arn"}
    },
    "AuthorizationException":{
      "type":"structure",
      "members":{
        "Message":{"shape":"Message"}
      },
      "exception":true
    },
    "AutomatedDiscoveryInformation":{
      "type":"structure",
      "members":{
        "LastRunTime":{"shape":"DateTime"}
      }
    },
    "Boolean":{"type":"boolean"},
    "BorrowConfiguration":{
      "type":"structure",
      "required":[
        "AllowEarlyCheckIn",
        "MaxTimeToLiveInMinutes"
      ],
      "members":{
        "AllowEarlyCheckIn":{"shape":"BoxBoolean"},
        "MaxTimeToLiveInMinutes":{"shape":"BoxInteger"}
      }
    },
    "BoxBoolean":{"type":"boolean"},
    "BoxInteger":{"type":"integer"},
    "BoxLong":{"type":"long"},
    "CheckInLicenseRequest":{
      "type":"structure",
      "required":["LicenseConsumptionToken"],
      "members":{
        "LicenseConsumptionToken":{"shape":"String"},
        "Beneficiary":{"shape":"String"}
      }
    },
    "CheckInLicenseResponse":{
```

```
   "type":"structure",
   "members":{
   }
  },
  "CheckoutBorrowLicenseRequest":{
   "type":"structure",
   "required":[
    "LicenseArn",
    "Entitlements",
    "DigitalSignatureMethod",
    "ClientToken"
   ],
   "members":{
    "LicenseArn":{"shape":"Arn"},
    "Entitlements":{"shape":"EntitlementDataList"},
    "DigitalSignatureMethod":{"shape":"DigitalSignatureMethod"},
    "NodeId":{"shape":"String"},
    "CheckoutMetadata":{"shape":"MetadataList"},
    "ClientToken":{"shape":"ClientToken"}
   }
  },

  "CheckoutBorrowLicenseResponse":{
   "type":"structure",
   "members":{
    "LicenseArn":{"shape":"Arn"},
    "LicenseConsumptionToken":{"shape":"String"},
    "EntitlementsAllowed":{"shape":"EntitlementDataList"},
    "NodeId":{"shape":"String"},
    "SignedToken":{"shape":"SignedToken"},
    "IssuedAt":{"shape":"ISO8601DateTime"},
    "Expiration":{"shape":"ISO8601DateTime"},
    "CheckoutMetadata":{"shape":"MetadataList"}
   }
  },
  "CheckoutLicenseRequest":{
   "type":"structure",
   "required":[
    "ProductSKU",
    "CheckoutType",
    "KeyFingerprint",
    "Entitlements",
    "ClientToken"
   ],
   "members":{
    "ProductSKU":{"shape":"String"},
    "CheckoutType":{"shape":"CheckoutType"},
    "KeyFingerprint":{"shape":"String"},
```

```
      "Entitlements":{"shape":"EntitlementDataList"},
      "ClientToken":{"shape":"ClientToken"},
      "Beneficiary":{"shape":"String"},
      "NodeId":{"shape":"String"}
    }
  },
  "CheckoutLicenseResponse":{
    "type":"structure",
    "members":{
      "CheckoutType":{"shape":"CheckoutType"},
      "LicenseConsumptionToken":{"shape":"String"},
      "EntitlementsAllowed":{"shape":"EntitlementDataList"},
      "SignedToken":{"shape":"SignedToken"},
      "NodeId":{"shape":"String"},
      "IssuedAt":{"shape":"ISO8601DateTime"},
      "Expiration":{"shape":"ISO8601DateTime"},
      "LicenseArn":{"shape":"String"}
    }
  },
  "CheckoutType":{
    "type":"string",
    "enum":[
      "PROVISIONAL",
      "PERPETUAL"
    ]
  },
  "ClientRequestToken":{
    "type":"string",
    "max":36,
    "min":1
  },
  "ClientToken":{
    "type":"string",
    "max":2048,
    "pattern":"\\S+"
  },
  "ConflictException":{
    "type":"structure",
    "members":{
      "Message":{"shape":"Message"}
    },
    "exception":true
  },
  "ConsumedLicenseSummary":{
    "type":"structure",
    "members":{
      "ResourceType":{"shape":"ResourceType"},
      "ConsumedLicenses":{"shape":"BoxLong"}
```

```
    }
  },
  "ConsumedLicenseSummaryList":{
   "type":"list",
   "member":{"shape":"ConsumedLicenseSummary"}
  },
  "ConsumptionConfiguration":{
   "type":"structure",
   "members":{
    "RenewType":{"shape":"RenewType"},
    "ProvisionalConfiguration":{"shape":"ProvisionalConfiguration"},
    "BorrowConfiguration":{"shape":"BorrowConfiguration"}
   }
  },
  "CreateGrantRequest":{
   "type":"structure",
   "required":[
    "ClientToken",
    "GrantName",
    "LicenseArn",
    "Principals",
    "HomeRegion",
    "AllowedOperations"
   ],
   "members":{
    "ClientToken":{"shape":"ClientToken"},
    "GrantName":{"shape":"String"},
    "LicenseArn":{"shape":"Arn"},
    "Principals":{"shape":"PrincipalArnList"},
    "HomeRegion":{"shape":"String"},
    "AllowedOperations":{"shape":"AllowedOperationList"}
   }
  },
  "CreateGrantResponse":{
   "type":"structure",
   "members":{
    "GrantArn":{"shape":"Arn"},
    "Status":{"shape":"GrantStatus"},
    "Version":{"shape":"String"}
   }
  },
  "CreateGrantVersionRequest":{
   "type":"structure",
   "required":[
    "ClientToken",
    "GrantArn"
   ],
   "members":{
```

```
    "ClientToken":{"shape":"ClientToken"},
    "GrantArn":{"shape":"Arn"},
    "GrantName":{"shape":"String"},
    "AllowedOperations":{"shape":"AllowedOperationList"},
    "Status":{"shape":"GrantStatus"},
    "StatusReason":{"shape":"StatusReasonMessage"},
    "SourceVersion":{"shape":"String"},
    "Options":{"shape":"Options"}
   }
  },
  "CreateGrantVersionResponse":{
   "type":"structure",
   "members":{
    "GrantArn":{"shape":"Arn"},
    "Status":{"shape":"GrantStatus"},
    "Version":{"shape":"String"}
   }
  },
  "CreateLicenseConfigurationRequest":{
   "type":"structure",
   "required":[
    "Name",
    "LicenseCountingType"
   ],
   "members":{
    "Name":{"shape":"String"},
    "Description":{"shape":"String"},
    "LicenseCountingType":{"shape":"LicenseCountingType"},
    "LicenseCount":{"shape":"BoxLong"},
    "LicenseCountHardLimit":{"shape":"BoxBoolean"},
    "LicenseRules":{"shape":"StringList"},
    "Tags":{"shape":"TagList"},
    "DisassociateWhenNotFound":{"shape":"BoxBoolean"},
    "ProductInformationList":{"shape":"ProductInformationList"}
   }
  },
  "CreateLicenseConfigurationResponse":{
   "type":"structure",
   "members":{
    "LicenseConfigurationArn":{"shape":"String"}
   }
  },
  "CreateLicenseConversionTaskForResourceRequest":{
   "type":"structure",
   "required":[
    "ResourceArn",
    "SourceLicenseContext",
    "DestinationLicenseContext"
```

        ],
      "members":{
        "ResourceArn":{"shape":"Arn"},
         "SourceLicenseContext":{"shape":"LicenseConversionContext"},
         "DestinationLicenseContext":{"shape":"LicenseConversionContext"}
       }
     },
    "CreateLicenseConversionTaskForResourceResponse":{
      "type":"structure",
      "members":{
        "LicenseConversionTaskId":{"shape":"LicenseConversionTaskId"}
       }
     },
    "CreateLicenseManagerReportGeneratorRequest":{
      "type":"structure",
      "required":[
        "ReportGeneratorName",
        "Type",
        "ReportContext",
        "ReportFrequency",
        "ClientToken"
       ],
      "members":{
        "ReportGeneratorName":{"shape":"ReportGeneratorName"},
        "Type":{"shape":"ReportTypeList"},
        "ReportContext":{"shape":"ReportContext"},
        "ReportFrequency":{"shape":"ReportFrequency"},
        "ClientToken":{"shape":"ClientRequestToken"},
        "Description":{"shape":"String"},
        "Tags":{"shape":"TagList"}
       }
     },
    "CreateLicenseManagerReportGeneratorResponse":{
      "type":"structure",
      "members":{
        "LicenseManagerReportGeneratorArn":{"shape":"String"}
       }
     },
    "CreateLicenseRequest":{
      "type":"structure",
      "required":[
        "LicenseName",
        "ProductName",
        "ProductSKU",
        "Issuer",
        "HomeRegion",
        "Validity",
        "Entitlements",

```
        "Beneficiary",
        "ConsumptionConfiguration",
        "ClientToken"
      ],
      "members":{
       "LicenseName":{"shape":"String"},
       "ProductName":{"shape":"String"},
       "ProductSKU":{"shape":"String"},
       "Issuer":{"shape":"Issuer"},
       "HomeRegion":{"shape":"String"},
       "Validity":{"shape":"DatetimeRange"},
       "Entitlements":{"shape":"EntitlementList"},
       "Beneficiary":{"shape":"String"},
       "ConsumptionConfiguration":{"shape":"ConsumptionConfiguration"},
       "LicenseMetadata":{"shape":"MetadataList"},
       "ClientToken":{"shape":"ClientToken"}

    }
   },
   "CreateLicenseResponse":{
    "type":"structure",
    "members":{
     "LicenseArn":{"shape":"Arn"},
     "Status":{"shape":"LicenseStatus"},
     "Version":{"shape":"String"}
    }
   },
   "CreateLicenseVersionRequest":{
    "type":"structure",
    "required":[
     "LicenseArn",
     "LicenseName",
     "ProductName",
     "Issuer",
     "HomeRegion",
     "Validity",
     "Entitlements",
     "ConsumptionConfiguration",
     "Status",
     "ClientToken"
    ],
    "members":{
     "LicenseArn":{"shape":"Arn"},
     "LicenseName":{"shape":"String"},
     "ProductName":{"shape":"String"},
     "Issuer":{"shape":"Issuer"},
     "HomeRegion":{"shape":"String"},
     "Validity":{"shape":"DatetimeRange"},
```

```
    "LicenseMetadata":{"shape":"MetadataList"},
    "Entitlements":{"shape":"EntitlementList"},
    "ConsumptionConfiguration":{"shape":"ConsumptionConfiguration"},
    "Status":{"shape":"LicenseStatus"},
    "ClientToken":{"shape":"ClientToken"},
    "SourceVersion":{"shape":"String"}
   }
  },
  "CreateLicenseVersionResponse":{
   "type":"structure",
   "members":{
    "LicenseArn":{"shape":"Arn"},
    "Version":{"shape":"String"},
    "Status":{"shape":"LicenseStatus"}
   }
  },
  "CreateTokenRequest":{
   "type":"structure",
   "required":[
    "LicenseArn",
    "ClientToken"
   ],
   "members":{
    "LicenseArn":{"shape":"Arn"},
    "RoleArns":{"shape":"ArnList"},
    "ExpirationInDays":{"shape":"Integer"},
    "TokenProperties":{"shape":"MaxSize3StringList"},
    "ClientToken":{"shape":"ClientToken"}
   }
  },
  "CreateTokenResponse":{
   "type":"structure",
   "members":{
    "TokenId":{"shape":"String"},
    "TokenType":{"shape":"TokenType"},
    "Token":{"shape":"TokenString"}
   }
  },
  "DateTime":{"type":"timestamp"},
  "DatetimeRange":{

"type":"structure",
   "required":["Begin"],
   "members":{
    "Begin":{"shape":"ISO8601DateTime"},
    "End":{"shape":"ISO8601DateTime"}
   }
  },
```

"DeleteGrantRequest":{
 "type":"structure",
 "required":[
  "GrantArn",
  "Version"
 ],
 "members":{
  "GrantArn":{"shape":"Arn"},
  "StatusReason":{"shape":"StatusReasonMessage"},
  "Version":{"shape":"String"}
 }
},
"DeleteGrantResponse":{
 "type":"structure",
 "members":{
  "GrantArn":{"shape":"Arn"},
  "Status":{"shape":"GrantStatus"},
  "Version":{"shape":"String"}
 }
},
"DeleteLicenseConfigurationRequest":{
 "type":"structure",
 "required":["LicenseConfigurationArn"],
 "members":{
  "LicenseConfigurationArn":{"shape":"String"}
 }
},
"DeleteLicenseConfigurationResponse":{
 "type":"structure",
 "members":{
 }
},
"DeleteLicenseManagerReportGeneratorRequest":{
 "type":"structure",
 "required":["LicenseManagerReportGeneratorArn"],
 "members":{
  "LicenseManagerReportGeneratorArn":{"shape":"String"}
 }
},
"DeleteLicenseManagerReportGeneratorResponse":{
 "type":"structure",
 "members":{
 }
},
"DeleteLicenseRequest":{
 "type":"structure",
 "required":[
  "LicenseArn",

```
    "SourceVersion"
  ],
  "members":{
    "LicenseArn":{"shape":"Arn"},
    "SourceVersion":{"shape":"String"}
  }
},
"DeleteLicenseResponse":{
  "type":"structure",
  "members":{
    "Status":{"shape":"LicenseDeletionStatus"},
    "DeletionDate":{"shape":"ISO8601DateTime"}
  }
},
"DeleteTokenRequest":{
  "type":"structure",
  "required":["TokenId"],
  "members":{
    "TokenId":{"shape":"String"}
  }
},
"DeleteTokenResponse":{
  "type":"structure",
  "members":{
  }
},
"DigitalSignatureMethod":{

  "type":"string",
  "enum":["JWT_PS384"]
},
"Entitlement":{
  "type":"structure",
  "required":[
    "Name",
    "Unit"
  ],
  "members":{
    "Name":{"shape":"String"},
    "Value":{"shape":"String"},
    "MaxCount":{"shape":"Long"},
    "Overage":{"shape":"BoxBoolean"},
    "Unit":{"shape":"EntitlementUnit"},
    "AllowCheckIn":{"shape":"BoxBoolean"}
  }
},
"EntitlementData":{
  "type":"structure",
```

```
   "required":[
    "Name",
    "Unit"
   ],
   "members":{
    "Name":{"shape":"String"},
    "Value":{"shape":"String"},
    "Unit":{"shape":"EntitlementDataUnit"}
   }
  },
  "EntitlementDataList":{
   "type":"list",
   "member":{"shape":"EntitlementData"}
  },
  "EntitlementDataUnit":{
   "type":"string",
   "enum":[
    "Count",
    "None",
    "Seconds",
    "Microseconds",
    "Milliseconds",
    "Bytes",
    "Kilobytes",

    "Megabytes",
    "Gigabytes",
    "Terabytes",
    "Bits",
    "Kilobits",
    "Megabits",
    "Gigabits",
    "Terabits",
    "Percent",
    "Bytes/Second",
    "Kilobytes/Second",
    "Megabytes/Second",
    "Gigabytes/Second",
    "Terabytes/Second",
    "Bits/Second",
    "Kilobits/Second",
    "Megabits/Second",
    "Gigabits/Second",
    "Terabits/Second",
    "Count/Second"
   ]
  },
  "EntitlementList":{
```

```
      "type":"list",
      "member":{"shape":"Entitlement"}
    },
    "EntitlementNotAllowedException":{
      "type":"structure",
      "members":{
        "Message":{"shape":"Message"}
      },
      "exception":true
    },
    "EntitlementUnit":{
      "type":"string",
      "enum":[
        "Count",
        "None",
        "Seconds",
        "Microseconds",
        "Milliseconds",
        "Bytes",
        "Kilobytes",
        "Megabytes",
        "Gigabytes",
        "Terabytes",

"Bits",
        "Kilobits",
        "Megabits",
        "Gigabits",
        "Terabits",
        "Percent",
        "Bytes/Second",
        "Kilobytes/Second",
        "Megabytes/Second",
        "Gigabytes/Second",
        "Terabytes/Second",
        "Bits/Second",
        "Kilobits/Second",
        "Megabits/Second",
        "Gigabits/Second",
        "Terabits/Second",
        "Count/Second"
      ]
    },
    "EntitlementUsage":{
      "type":"structure",
      "required":[
        "Name",
        "ConsumedValue",
```

```
      "Unit"
     ],
     "members":{
      "Name":{"shape":"String"},
      "ConsumedValue":{"shape":"String"},
      "MaxCount":{"shape":"String"},
      "Unit":{"shape":"EntitlementDataUnit"}
     }
    },
    "EntitlementUsageList":{
     "type":"list",
     "member":{"shape":"EntitlementUsage"}
    },
    "ExtendLicenseConsumptionRequest":{
     "type":"structure",
     "required":["LicenseConsumptionToken"],
     "members":{
      "LicenseConsumptionToken":{"shape":"String"},
      "DryRun":{"shape":"Boolean"}
     }
    },
    "ExtendLicenseConsumptionResponse":{
     "type":"structure",
     "members":{
      "LicenseConsumptionToken":{"shape":"String"},
      "Expiration":{"shape":"ISO8601DateTime"}
     }
    },
    "FailedDependencyException":{
     "type":"structure",
     "members":{
      "Message":{"shape":"Message"},
      "ErrorCode":{"shape":"String"}
     },
     "exception":true
    },
    "Filter":{
     "type":"structure",
     "members":{
      "Name":{"shape":"FilterName"},
      "Values":{"shape":"FilterValues"}
     }
    },
    "FilterLimitExceededException":{
     "type":"structure",
     "members":{
      "Message":{"shape":"Message"}
     },
```

```
  "exception":true
},
"FilterList":{
 "type":"list",
 "member":{"shape":"Filter"}
},
"FilterName":{"type":"string"},
"FilterValue":{"type":"string"},
"FilterValues":{
 "type":"list",
 "member":{"shape":"FilterValue"}
},
"Filters":{
 "type":"list",
 "member":{"shape":"Filter"}
},
"GetAccessTokenRequest":{
 "type":"structure",
 "required":["Token"],
 "members":{
  "Token":{"shape":"TokenString"},
  "TokenProperties":{"shape":"MaxSize3StringList"}
 }
},
"GetAccessTokenResponse":{
 "type":"structure",
 "members":{
  "AccessToken":{"shape":"TokenString"}
 }
},
"GetGrantRequest":{
 "type":"structure",
 "required":["GrantArn"],
 "members":{
  "GrantArn":{"shape":"Arn"},
  "Version":{"shape":"String"}
 }
},
"GetGrantResponse":{
 "type":"structure",
 "members":{
  "Grant":{"shape":"Grant"}
 }
},
"GetLicenseConfigurationRequest":{
 "type":"structure",
 "required":["LicenseConfigurationArn"],
 "members":{
```

      "LicenseConfigurationArn":{"shape":"String"}
     }
    },
    "GetLicenseConfigurationResponse":{
     "type":"structure",
     "members":{
      "LicenseConfigurationId":{"shape":"String"},
      "LicenseConfigurationArn":{"shape":"String"},
      "Name":{"shape":"String"},
      "Description":{"shape":"String"},
      "LicenseCountingType":{"shape":"LicenseCountingType"},
      "LicenseRules":{"shape":"StringList"},
      "LicenseCount":{"shape":"BoxLong"},
      "LicenseCountHardLimit":{"shape":"BoxBoolean"},
      "ConsumedLicenses":{"shape":"BoxLong"},
      "Status":{"shape":"String"},
      "OwnerAccountId":{"shape":"String"},
      "ConsumedLicenseSummaryList":{"shape":"ConsumedLicenseSummaryList"},
      "ManagedResourceSummaryList":{"shape":"ManagedResourceSummaryList"},
      "Tags":{"shape":"TagList"},
      "ProductInformationList":{"shape":"ProductInformationList"},
      "AutomatedDiscoveryInformation":{"shape":"AutomatedDiscoveryInformation"},
      "DisassociateWhenNotFound":{"shape":"BoxBoolean"}
     }
    },
    "GetLicenseConversionTaskRequest":{
     "type":"structure",
     "required":["LicenseConversionTaskId"],
     "members":{
      "LicenseConversionTaskId":{"shape":"LicenseConversionTaskId"}
     }
    },
    "GetLicenseConversionTaskResponse":{
     "type":"structure",
     "members":{
      "LicenseConversionTaskId":{"shape":"LicenseConversionTaskId"},
      "ResourceArn":{"shape":"String"},
      "SourceLicenseContext":{"shape":"LicenseConversionContext"},
      "DestinationLicenseContext":{"shape":"LicenseConversionContext"},
      "StatusMessage":{"shape":"String"},
      "Status":{"shape":"LicenseConversionTaskStatus"},
      "StartTime":{"shape":"DateTime"},
      "LicenseConversionTime":{"shape":"DateTime"},
      "EndTime":{"shape":"DateTime"}
     }
    },
    "GetLicenseManagerReportGeneratorRequest":{
     "type":"structure",

```
   "required":["LicenseManagerReportGeneratorArn"],
   "members":{
     "LicenseManagerReportGeneratorArn":{"shape":"String"}
    }
  },
  "GetLicenseManagerReportGeneratorResponse":{
   "type":"structure",
   "members":{
     "ReportGenerator":{"shape":"ReportGenerator"}
    }
  },
  "GetLicenseRequest":{
   "type":"structure",
   "required":["LicenseArn"],
   "members":{
     "LicenseArn":{"shape":"Arn"},
     "Version":{"shape":"String"}
    }
  },
  "GetLicenseResponse":{
   "type":"structure",
   "members":{
     "License":{"shape":"License"}
    }
  },
  "GetLicenseUsageRequest":{
   "type":"structure",
   "required":["LicenseArn"],
   "members":{
     "LicenseArn":{"shape":"Arn"}
    }
  },
  "GetLicenseUsageResponse":{
   "type":"structure",
   "members":{
     "LicenseUsage":{"shape":"LicenseUsage"}
    }
  },
  "GetServiceSettingsRequest":{
   "type":"structure",
   "members":{
    }
  },
  "GetServiceSettingsResponse":{
   "type":"structure",
   "members":{
     "S3BucketArn":{"shape":"String"},
```

```
      "SnsTopicArn":{"shape":"String"},
     "OrganizationConfiguration":{"shape":"OrganizationConfiguration"},
     "EnableCrossAccountsDiscovery":{"shape":"BoxBoolean"},
     "LicenseManagerResourceShareArn":{"shape":"String"}
    }
   },
   "Grant":{
    "type":"structure",
    "required":[
     "GrantArn",
     "GrantName",
     "ParentArn",
     "LicenseArn",
     "GranteePrincipalArn",
     "HomeRegion",
     "GrantStatus",
     "Version",
     "GrantedOperations"
    ],
    "members":{
     "GrantArn":{"shape":"Arn"},
     "GrantName":{"shape":"String"},
     "ParentArn":{"shape":"Arn"},
     "LicenseArn":{"shape":"Arn"},
     "GranteePrincipalArn":{"shape":"Arn"},
     "HomeRegion":{"shape":"String"},
     "GrantStatus":{"shape":"GrantStatus"},
     "StatusReason":{"shape":"StatusReasonMessage"},
     "Version":{"shape":"String"},
     "GrantedOperations":{"shape":"AllowedOperationList"},
      "Options":{"shape":"Options"}
    }
   },
   "GrantList":{
    "type":"list",
    "member":{"shape":"Grant"}
   },
   "GrantStatus":{
    "type":"string",
    "enum":[
     "PENDING_WORKFLOW",
     "PENDING_ACCEPT",
     "REJECTED",
     "ACTIVE",
     "FAILED_WORKFLOW",
     "DELETED",
     "PENDING_DELETE",
     "DISABLED",
```

```
    "WORKFLOW_COMPLETED"
   ]
  },
  "GrantedLicense":{
   "type":"structure",
   "members":{
    "LicenseArn":{"shape":"Arn"},
    "LicenseName":{"shape":"String"},
    "ProductName":{"shape":"String"},
    "ProductSKU":{"shape":"String"},
    "Issuer":{"shape":"IssuerDetails"},
    "HomeRegion":{"shape":"String"},
    "Status":{"shape":"LicenseStatus"},
    "Validity":{"shape":"DatetimeRange"},
    "Beneficiary":{"shape":"String"},
    "Entitlements":{"shape":"EntitlementList"},
    "ConsumptionConfiguration":{"shape":"ConsumptionConfiguration"},

  "LicenseMetadata":{"shape":"MetadataList"},
    "CreateTime":{"shape":"ISO8601DateTime"},
    "Version":{"shape":"String"},
    "ReceivedMetadata":{"shape":"ReceivedMetadata"}
   }
  },
  "GrantedLicenseList":{
   "type":"list",
   "member":{"shape":"GrantedLicense"}
  },
  "ISO8601DateTime":{
   "type":"string",
   "max":50,
   "pattern":"^(-?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|[0-1][0-9]):([0-5][0-9]):([0-5][0-9])(\\.[0-9]+)?(Z|[+-](?:2[ 0-3]|[0-1][0-9]):[0-5][0-9])+$"
  },
  "Integer":{"type":"integer"},
  "InvalidParameterValueException":{
   "type":"structure",
   "members":{
    "Message":{"shape":"Message"}
   },
   "exception":true,
   "synthetic":true
  },
  "InvalidResourceStateException":{
   "type":"structure",
   "members":{
    "Message":{"shape":"Message"}
   },
```

```
    "exception":true
  },
  "InventoryFilter":{
   "type":"structure",
   "required":[
     "Name",
     "Condition"
   ],
   "members":{
     "Name":{"shape":"String"},
     "Condition":{"shape":"InventoryFilterCondition"},
     "Value":{"shape":"String"}
   }
  },
  "InventoryFilterCondition":{
   "type":"string",
   "enum":[
     "EQUALS",
     "NOT_EQUALS",
     "BEGINS_WITH",
     "CONTAINS"
   ]
  },
  "InventoryFilterList":{
   "type":"list",
   "member":{"shape":"InventoryFilter"}
  },
  "Issuer":{
   "type":"structure",
   "required":["Name"],
   "members":{
     "Name":{"shape":"String"},
     "SignKey":{"shape":"String"}
   }
  },
  "IssuerDetails":{
   "type":"structure",
   "members":{
     "Name":{"shape":"String"},
     "SignKey":{"shape":"String"},
     "KeyFingerprint":{"shape":"String"}
   }
  },
  "License":{
   "type":"structure",
   "members":{
     "LicenseArn":{"shape":"Arn"},
     "LicenseName":{"shape":"String"},
```

```
"ProductName":{"shape":"String"},
  "ProductSKU":{"shape":"String"},
  "Issuer":{"shape":"IssuerDetails"},
  "HomeRegion":{"shape":"String"},
  "Status":{"shape":"LicenseStatus"},
  "Validity":{"shape":"DatetimeRange"},
  "Beneficiary":{"shape":"String"},
  "Entitlements":{"shape":"EntitlementList"},
  "ConsumptionConfiguration":{"shape":"ConsumptionConfiguration"},
  "LicenseMetadata":{"shape":"MetadataList"},
  "CreateTime":{"shape":"ISO8601DateTime"},
  "Version":{"shape":"String"}
 }
},
"LicenseConfiguration":{
 "type":"structure",
 "members":{
  "LicenseConfigurationId":{"shape":"String"},
  "LicenseConfigurationArn":{"shape":"String"},
  "Name":{"shape":"String"},
  "Description":{"shape":"String"},
  "LicenseCountingType":{"shape":"LicenseCountingType"},
  "LicenseRules":{"shape":"StringList"},
  "LicenseCount":{"shape":"BoxLong"},
  "LicenseCountHardLimit":{"shape":"BoxBoolean"},
   "DisassociateWhenNotFound":{"shape":"BoxBoolean"},
  "ConsumedLicenses":{"shape":"BoxLong"},
  "Status":{"shape":"String"},
  "OwnerAccountId":{"shape":"String"},
  "ConsumedLicenseSummaryList":{"shape":"ConsumedLicenseSummaryList"},
  "ManagedResourceSummaryList":{"shape":"ManagedResourceSummaryList"},
  "ProductInformationList":{"shape":"ProductInformationList"},
  "AutomatedDiscoveryInformation":{"shape":"AutomatedDiscoveryInformation"}
 }
},
"LicenseConfigurationAssociation":{
 "type":"structure",
 "members":{
  "ResourceArn":{"shape":"String"},
  "ResourceType":{"shape":"ResourceType"},
  "ResourceOwnerId":{"shape":"String"},
  "AssociationTime":{"shape":"DateTime"},
  "AmiAssociationScope":{"shape":"String"}
 }
},
"LicenseConfigurationAssociations":{
 "type":"list",
```

```
      "member":{"shape":"LicenseConfigurationAssociation"}
    },
    "LicenseConfigurationStatus":{
     "type":"string",
     "enum":[
       "AVAILABLE",
       "DISABLED"
     ]
    },
    "LicenseConfigurationUsage":{
     "type":"structure",
     "members":{
       "ResourceArn":{"shape":"String"},
       "ResourceType":{"shape":"ResourceType"},
       "ResourceStatus":{"shape":"String"},
       "ResourceOwnerId":{"shape":"String"},
       "AssociationTime":{"shape":"DateTime"},
       "ConsumedLicenses":{"shape":"BoxLong"}
     }
    },
    "LicenseConfigurationUsageList":{
     "type":"list",
     "member":{"shape":"LicenseConfigurationUsage"}
    },
    "LicenseConfigurations":{
     "type":"list",
     "member":{"shape":"LicenseConfiguration"}
    },
    "LicenseConversionContext":{
     "type":"structure",
     "members":{
       "UsageOperation":{"shape":"UsageOperation"}
     }
    },
    "LicenseConversionTask":{
     "type":"structure",
     "members":{
       "LicenseConversionTaskId":{"shape":"LicenseConversionTaskId"},
        "ResourceArn":{"shape":"String"},
       "SourceLicenseContext":{"shape":"LicenseConversionContext"},
       "DestinationLicenseContext":{"shape":"LicenseConversionContext"},
       "Status":{"shape":"LicenseConversionTaskStatus"},
       "StatusMessage":{"shape":"String"},
       "StartTime":{"shape":"DateTime"},
       "LicenseConversionTime":{"shape":"DateTime"},
       "EndTime":{"shape":"DateTime"}
     }
    },
```

```
"LicenseConversionTaskId":{
 "type":"string",
 "max":50,
 "pattern":"^lct-[a-zA-Z0-9]*"
},
"LicenseConversionTaskStatus":{
 "type":"string",
 "enum":[
  "IN_PROGRESS",
  "SUCCEEDED",
  "FAILED"
 ]
},
"LicenseConversionTasks":{
 "type":"list",
 "member":{"shape":"LicenseConversionTask"}
},
"LicenseCountingType":{
 "type":"string",
 "enum":[
  "vCPU",
  "Instance",
  "Core",
  "Socket"
 ]
},
"LicenseDeletionStatus":{
 "type":"string",
 "enum":[
  "PENDING_DELETE",
  "DELETED"
 ]
},
"LicenseList":{
 "type":"list",
 "member":{"shape":"License"}
},
"LicenseOperationFailure":{
 "type":"structure",
 "members":{
  "ResourceArn":{"shape":"String"},
  "ResourceType":{"shape":"ResourceType"},
  "ErrorMessage":{"shape":"String"},
  "FailureTime":{"shape":"DateTime"},
  "OperationName":{"shape":"String"},
  "ResourceOwnerId":{"shape":"String"},
  "OperationRequestedBy":{"shape":"String"},
  "MetadataList":{"shape":"MetadataList"}
```

```
      }
    },
    "LicenseOperationFailureList":{
      "type":"list",
      "member":{"shape":"LicenseOperationFailure"}
    },
    "LicenseSpecification":{
      "type":"structure",
      "required":["LicenseConfigurationArn"],
      "members":{
        "LicenseConfigurationArn":{"shape":"String"},
        "AmiAssociationScope":{"shape":"String"}
      }
    },
    "LicenseSpecifications":{
      "type":"list",
      "member":{"shape":"LicenseSpecification"}
    },
    "LicenseStatus":{
      "type":"string",
      "enum":[
        "AVAILABLE",
        "PENDING_AVAILABLE",
        "DEACTIVATED",
        "SUSPENDED",
        "EXPIRED",
        "PENDING_DELETE",
        "DELETED"
      ]
    },
    "LicenseUsage":{
      "type":"structure",
      "members":{
        "EntitlementUsages":{"shape":"EntitlementUsageList"}
      }
    },
    "LicenseUsageException":{
      "type":"structure",
      "members":{
        "Message":{"shape":"Message"}
      },
      "exception":true
    },
    "ListAssociationsForLicenseConfigurationRequest":{
      "type":"structure",
      "required":["LicenseConfigurationArn"],
      "members":{
        "LicenseConfigurationArn":{"shape":"String"},
```

```
     "MaxResults":{"shape":"BoxInteger"},
     "NextToken":{"shape":"String"}
    }
   },
   "ListAssociationsForLicenseConfigurationResponse":{
    "type":"structure",
    "members":{
     "LicenseConfigurationAssociations":{"shape":"LicenseConfigurationAssociations"},
     "NextToken":{"shape":"String"}
    }
   },
   "ListDistributedGrantsRequest":{
    "type":"structure",
    "members":{
     "GrantArns":{"shape":"ArnList"},
     "Filters":{"shape":"FilterList"},
     "NextToken":{"shape":"String"},
     "MaxResults":{"shape":"MaxSize100"}
    }
   },
   "ListDistributedGrantsResponse":{
    "type":"structure",
    "members":{
     "Grants":{"shape":"GrantList"},
     "NextToken":{"shape":"String"}
    }
   },
   "ListFailuresForLicenseConfigurationOperationsRequest":{
    "type":"structure",
    "required":["LicenseConfigurationArn"],
    "members":{
     "LicenseConfigurationArn":{"shape":"String"},
     "MaxResults":{"shape":"BoxInteger"},
     "NextToken":{"shape":"String"}
    }
   },
   "ListFailuresForLicenseConfigurationOperationsResponse":{
    "type":"structure",
    "members":{
     "LicenseOperationFailureList":{"shape":"LicenseOperationFailureList"},
     "NextToken":{"shape":"String"}
    }
   },
   "ListLicenseConfigurationsRequest":{
    "type":"structure",
    "members":{
     "LicenseConfigurationArns":{"shape":"StringList"},
     "MaxResults":{"shape":"BoxInteger"},
```

```
      "NextToken":{"shape":"String"},
      "Filters":{"shape":"Filters"}
    }
  },
  "ListLicenseConfigurationsResponse":{
   "type":"structure",
   "members":{
     "LicenseConfigurations":{"shape":"LicenseConfigurations"},
     "NextToken":{"shape":"String"}
   }
  },
  "ListLicenseConversionTasksRequest":{
   "type":"structure",
   "members":{
     "NextToken":{"shape":"String"},
     "MaxResults":{"shape":"BoxInteger"},
     "Filters":{"shape":"Filters"}
   }
  },
  "ListLicenseConversionTasksResponse":{
   "type":"structure",
   "members":{
     "LicenseConversionTasks":{"shape":"LicenseConversionTasks"},
      "NextToken":{"shape":"String"}
   }
  },
  "ListLicenseManagerReportGeneratorsRequest":{
   "type":"structure",
   "members":{
     "Filters":{"shape":"FilterList"},
     "NextToken":{"shape":"String"},
     "MaxResults":{"shape":"MaxSize100"}
   }
  },
  "ListLicenseManagerReportGeneratorsResponse":{
   "type":"structure",
   "members":{
     "ReportGenerators":{"shape":"ReportGeneratorList"},
     "NextToken":{"shape":"String"}
   }
  },
  "ListLicenseSpecificationsForResourceRequest":{
   "type":"structure",
   "required":["ResourceArn"],
   "members":{
     "ResourceArn":{"shape":"String"},
     "MaxResults":{"shape":"BoxInteger"},
     "NextToken":{"shape":"String"}
```

```
      }
    },
    "ListLicenseSpecificationsForResourceResponse":{
      "type":"structure",
      "members":{
        "LicenseSpecifications":{"shape":"LicenseSpecifications"},
        "NextToken":{"shape":"String"}

}
    },
    "ListLicenseVersionsRequest":{
      "type":"structure",
      "required":["LicenseArn"],
      "members":{
        "LicenseArn":{"shape":"Arn"},
        "NextToken":{"shape":"String"},
        "MaxResults":{"shape":"MaxSize100"}
      }
    },
    "ListLicenseVersionsResponse":{
      "type":"structure",
      "members":{
        "Licenses":{"shape":"LicenseList"},
        "NextToken":{"shape":"String"}
      }
    },
    "ListLicensesRequest":{
      "type":"structure",
      "members":{
        "LicenseArns":{"shape":"ArnList"},
        "Filters":{"shape":"FilterList"},
        "NextToken":{"shape":"String"},
        "MaxResults":{"shape":"MaxSize100"}
      }
    },
    "ListLicensesResponse":{
      "type":"structure",
      "members":{
        "Licenses":{"shape":"LicenseList"},
        "NextToken":{"shape":"String"}
      }
    },
    "ListReceivedGrantsForOrganizationRequest":{
      "type":"structure",
      "required":["LicenseArn"],
      "members":{
        "LicenseArn":{"shape":"Arn"},
```

```
    "Filters":{"shape":"FilterList"},
    "NextToken":{"shape":"String"},
    "MaxResults":{"shape":"MaxSize100"}
   }
  },
  "ListReceivedGrantsForOrganizationResponse":{
   "type":"structure",
   "members":{
    "Grants":{"shape":"GrantList"},
    "NextToken":{"shape":"String"}
   }
  },
  "ListReceivedGrantsRequest":{
   "type":"structure",
   "members":{
    "GrantArns":{"shape":"ArnList"},
    "Filters":{"shape":"FilterList"},
    "NextToken":{"shape":"String"},
    "MaxResults":{"shape":"MaxSize100"}
   }
  },
  "ListReceivedGrantsResponse":{
   "type":"structure",
   "members":{
    "Grants":{"shape":"GrantList"},
    "NextToken":{"shape":"String"}
   }
  },
  "ListReceivedLicensesForOrganizationRequest":{
   "type":"structure",
   "members":{
    "Filters":{"shape":"FilterList"},
    "NextToken":{"shape":"String"},
    "MaxResults":{"shape":"MaxSize100"}
   }
  },
  "ListReceivedLicensesForOrganizationResponse":{
   "type":"structure",
   "members":{
    "Licenses":{"shape":"GrantedLicenseList"},
    "NextToken":{"shape":"String"}
   }
  },
  "ListReceivedLicensesRequest":{
   "type":"structure",
   "members":{
    "LicenseArns":{"shape":"ArnList"},
    "Filters":{"shape":"FilterList"},
```

```
    "NextToken":{"shape":"String"},
    "MaxResults":{"shape":"MaxSize100"}
   }
  },
  "ListReceivedLicensesResponse":{
   "type":"structure",
   "members":{
    "Licenses":{"shape":"GrantedLicenseList"},
    "NextToken":{"shape":"String"}
   }
  },
  "ListResourceInventoryRequest":{
   "type":"structure",
   "members":{
    "MaxResults":{"shape":"BoxInteger"},
    "NextToken":{"shape":"String"},
    "Filters":{"shape":"InventoryFilterList"}
   }
  },
  "ListResourceInventoryResponse":{
   "type":"structure",
   "members":{
    "ResourceInventoryList":{"shape":"ResourceInventoryList"},
    "NextToken":{"shape":"String"}
   }
  },
  "ListTagsForResourceRequest":{
   "type":"structure",
   "required":["ResourceArn"],
   "members":{
    "ResourceArn":{"shape":"String"}
   }
  },
  "ListTagsForResourceResponse":{
   "type":"structure",
   "members":{
    "Tags":{"shape":"TagList"}
   }
  },
  "ListTokensRequest":{
   "type":"structure",
   "members":{
    "TokenIds":{"shape":"StringList"},
    "Filters":{"shape":"FilterList"},
    "NextToken":{"shape":"String"},
    "MaxResults":{"shape":"MaxSize100"}
   }
  },
```

```
"ListTokensResponse":{
 "type":"structure",
 "members":{
  "Tokens":{"shape":"TokenList"},
  "NextToken":{"shape":"String"}
 }
},
"ListUsageForLicenseConfigurationRequest":{
 "type":"structure",
 "required":["LicenseConfigurationArn"],
 "members":{
  "LicenseConfigurationArn":{"shape":"String"},
   "MaxResults":{"shape":"BoxInteger"},
  "NextToken":{"shape":"String"},
  "Filters":{"shape":"Filters"}
 }
},
"ListUsageForLicenseConfigurationResponse":{
 "type":"structure",
 "members":{
  "LicenseConfigurationUsageList":{"shape":"LicenseConfigurationUsageList"},
  "NextToken":{"shape":"String"}
 }
},
"Location":{"type":"string"},
"Long":{"type":"long"},
"ManagedResourceSummary":{
 "type":"structure",
 "members":{
  "ResourceType":{"shape":"ResourceType"},
  "AssociationCount":{"shape":"BoxLong"}
 }
},
"ManagedResourceSummaryList":{
 "type":"list",
 "member":{"shape":"ManagedResourceSummary"}
},
"MaxSize100":{
 "type":"integer",
 "max":100,
 "min":1
},
"MaxSize3StringList":{
 "type":"list",
 "member":{"shape":"String"},
 "max":3
},
"Message":{"type":"string"},
```

```json
"Metadata":{
 "type":"structure",
  "members":{
   "Name":{"shape":"String"},
   "Value":{"shape":"String"}
  }
},
"MetadataList":{
 "type":"list",
 "member":{"shape":"Metadata"}
},
"NoEntitlementsAllowedException":{
 "type":"structure",
 "members":{
  "Message":{"shape":"Message"}
 },
 "exception":true
},
"Options":{
 "type":"structure",
 "members":{
  "ActivationOverrideBehavior":{"shape":"ActivationOverrideBehavior"}
 }
},
"OrganizationConfiguration":{
 "type":"structure",
 "required":["EnableIntegration"],
 "members":{
  "EnableIntegration":{"shape":"Boolean"}
 }
},
"PrincipalArnList":{
 "type":"list",
 "member":{"shape":"Arn"},
 "max":1,
 "min":1
},
"ProductInformation":{
 "type":"structure",
 "required":[
  "ResourceType",
  "ProductInformationFilterList"
 ],
 "members":{
  "ResourceType":{"shape":"String"},
  "ProductInformationFilterList":{"shape":"ProductInformationFilterList"}
 }
},
```

"ProductInformationFilter":{
 "type":"structure",
 "required":[
   "ProductInformationFilterName",
   "ProductInformationFilterComparator"
 ],
 "members":{
   "ProductInformationFilterName":{"shape":"String"},
   "ProductInformationFilterValue":{"shape":"StringList"},
   "ProductInformationFilterComparator":{"shape":"String"}
 }
},
"ProductInformationFilterList":{
 "type":"list",
 "member":{"shape":"ProductInformationFilter"}
},
"ProductInformationList":{
 "type":"list",
 "member":{"shape":"ProductInformation"}
},
"ProvisionalConfiguration":{
 "type":"structure",
 "required":["MaxTimeToLiveInMinutes"],
 "members":{
   "MaxTimeToLiveInMinutes":{"shape":"BoxInteger"}
 }
},
"RateLimitExceededException":{
 "type":"structure",
 "members":{
   "Message":{"shape":"Message"}
 },
 "exception":true
},
"ReceivedMetadata":{
 "type":"structure",
 "members":{
   "ReceivedStatus":{"shape":"ReceivedStatus"},
   "ReceivedStatusReason":{"shape":"StatusReasonMessage"},
   "AllowedOperations":{"shape":"AllowedOperationList"}
 }
},
"ReceivedStatus":{
 "type":"string",
 "enum":[
   "PENDING_WORKFLOW",
   "PENDING_ACCEPT",
   "REJECTED",

```
      "ACTIVE",
      "FAILED_WORKFLOW",
      "DELETED",
      "DISABLED",
      "WORKFLOW_COMPLETED"
     ]
    },
    "RedirectException":{
     "type":"structure",
     "members":{
      "Location":{"shape":"Location"},
      "Message":{"shape":"Message"}
     },
     "exception":true
    },
    "RejectGrantRequest":{
     "type":"structure",
     "required":["GrantArn"],
     "members":{
      "GrantArn":{"shape":"Arn"}
     }
    },
    "RejectGrantResponse":{
     "type":"structure",
     "members":{
      "GrantArn":{"shape":"Arn"},
      "Status":{"shape":"GrantStatus"},
      "Version":{"shape":"String"}
     }
    },
    "RenewType":{
     "type":"string",
     "enum":[
      "None",
      "Weekly",
      "Monthly"
     ]
    },
    "ReportContext":{
     "type":"structure",
     "required":["licenseConfigurationArns"],
     "members":{
      "licenseConfigurationArns":{"shape":"ArnList"}
     }
    },
    "ReportFrequency":{
     "type":"structure",
     "members":{
```

```
    "value":{"shape":"Integer"},
    "period":{"shape":"ReportFrequencyType"}
  }
},
"ReportFrequencyType":{
 "type":"string",
 "enum":[
   "DAY",
   "WEEK",
   "MONTH"
 ]
},
"ReportGenerator":{
 "type":"structure",
 "members":{
   "ReportGeneratorName":{"shape":"String"},
   "ReportType":{"shape":"ReportTypeList"},
   "ReportContext":{"shape":"ReportContext"},
   "ReportFrequency":{"shape":"ReportFrequency"},
    "LicenseManagerReportGeneratorArn":{"shape":"String"},
   "LastRunStatus":{"shape":"String"},
   "LastRunFailureReason":{"shape":"String"},
   "LastReportGenerationTime":{"shape":"String"},
   "ReportCreatorAccount":{"shape":"String"},
   "Description":{"shape":"String"},
   "S3Location":{"shape":"S3Location"},
   "CreateTime":{"shape":"String"},
   "Tags":{"shape":"TagList"}
  }
},
"ReportGeneratorList":{
 "type":"list",
 "member":{"shape":"ReportGenerator"}
},
"ReportGeneratorName":{
 "type":"string",
 "max":100,
 "min":1
},
"ReportType":{
 "type":"string",
 "enum":[
   "LicenseConfigurationSummaryReport",
   "LicenseConfigurationUsageReport"
 ]
},
"ReportTypeList":{
 "type":"list",
```

```
    "member":{"shape":"ReportType"}
   },
   "ResourceInventory":{
    "type":"structure",
    "members":{
     "ResourceId":{"shape":"String"},
      "ResourceType":{"shape":"ResourceType"},
     "ResourceArn":{"shape":"String"},
     "Platform":{"shape":"String"},
     "PlatformVersion":{"shape":"String"},
     "ResourceOwningAccountId":{"shape":"String"}
    }
   },
   "ResourceInventoryList":{
    "type":"list",
    "member":{"shape":"ResourceInventory"}
   },
   "ResourceLimitExceededException":{
    "type":"structure",
    "members":{
     "Message":{"shape":"Message"}
    },
    "exception":true
   },
   "ResourceNotFoundException":{
    "type":"structure",
    "members":{
     "Message":{"shape":"Message"}
    },
    "exception":true
   },
   "ResourceType":{
    "type":"string",
    "enum":[
     "EC2_INSTANCE",
     "EC2_HOST",
     "EC2_AMI",
     "RDS",
     "SYSTEMS_MANAGER_MANAGED_INSTANCE"
    ]
   },
   "S3Location":{
    "type":"structure",
    "members":{
     "bucket":{"shape":"String"},
     "keyPrefix":{"shape":"String"}
```

```
      }
    },
    "ServerInternalException":{
     "type":"structure",
     "members":{
       "Message":{"shape":"Message"}
     },
     "exception":true,
     "fault":true
    },
    "SignedToken":{
     "type":"string",
     "min":4096
    },
    "StatusReasonMessage":{
     "type":"string",
     "max":400,
     "pattern":"[\\s\\S]+"
    },
    "String":{"type":"string"},
    "StringList":{
     "type":"list",
     "member":{"shape":"String"}
    },
    "Tag":{
     "type":"structure",
     "members":{
       "Key":{"shape":"String"},
       "Value":{"shape":"String"}
     }
    },
    "TagKeyList":{
     "type":"list",
     "member":{"shape":"String"}
    },
    "TagList":{
     "type":"list",
     "member":{"shape":"Tag"}
    },
    "TagResourceRequest":{
     "type":"structure",
     "required":[
       "ResourceArn",
       "Tags"
     ],
     "members":{
       "ResourceArn":{"shape":"String"},
       "Tags":{"shape":"TagList"}
```

```
      }
    },
    "TagResourceResponse":{
      "type":"structure",
      "members":{
      }
    },
    "TokenData":{
      "type":"structure",
      "members":{
        "TokenId":{"shape":"String"},
        "TokenType":{"shape":"String"},
        "LicenseArn":{"shape":"String"},
        "ExpirationTime":{"shape":"ISO8601DateTime"},
        "TokenProperties":{"shape":"MaxSize3StringList"},
        "RoleArns":{"shape":"ArnList"},
        "Status":{"shape":"String"}
      }
    },
    "TokenList":{
      "type":"list",
      "member":{"shape":"TokenData"}
    },
    "TokenString":{
      "type":"string",
      "max":4096,
      "pattern":"\\S+"
    },
    "TokenType":{
      "type":"string",
      "enum":["REFRESH_TOKEN"]
    },
    "UnsupportedDigitalSignatureMethodException":{
      "type":"structure",
      "members":{
        "Message":{"shape":"Message"}
      },
      "exception":true
    },
    "UntagResourceRequest":{
      "type":"structure",
      "required":[
        "ResourceArn",
        "TagKeys"
      ],
      "members":{
        "ResourceArn":{"shape":"String"},
        "TagKeys":{"shape":"TagKeyList"}
```

```
    }
  },
  "UntagResourceResponse":{
   "type":"structure",
   "members":{
   }
  },
  "UpdateLicenseConfigurationRequest":{
   "type":"structure",
   "required":["LicenseConfigurationArn"],
   "members":{
    "LicenseConfigurationArn":{"shape":"String"},
    "LicenseConfigurationStatus":{"shape":"LicenseConfigurationStatus"},
    "LicenseRules":{"shape":"StringList"},
    "LicenseCount":{"shape":"BoxLong"},
    "LicenseCountHardLimit":{"shape":"BoxBoolean"},
    "Name":{"shape":"String"},
    "Description":{"shape":"String"},
    "ProductInformationList":{"shape":"ProductInformationList"},
    "DisassociateWhenNotFound":{"shape":"BoxBoolean"}
   }
  },
  "UpdateLicenseConfigurationResponse":{
   "type":"structure",
   "members":{
   }
  },
  "UpdateLicenseManagerReportGeneratorRequest":{
   "type":"structure",
   "required":[
    "LicenseManagerReportGeneratorArn",
    "ReportGeneratorName",
    "Type",
    "ReportContext",
    "ReportFrequency",
    "ClientToken"
   ],
   "members":{
    "LicenseManagerReportGeneratorArn":{"shape":"String"},
    "ReportGeneratorName":{"shape":"ReportGeneratorName"},
    "Type":{"shape":"ReportTypeList"},
    "ReportContext":{"shape":"ReportContext"},
    "ReportFrequency":{"shape":"ReportFrequency"},
    "ClientToken":{"shape":"ClientRequestToken"},
    "Description":{"shape":"String"}
   }
  },
  "UpdateLicenseManagerReportGeneratorResponse":{
```

```
      "type":"structure",
      "members":{
      }
    },
    "UpdateLicenseSpecificationsForResourceRequest":{
      "type":"structure",
      "required":["ResourceArn"],
      "members":{
        "ResourceArn":{"shape":"String"},
        "AddLicenseSpecifications":{"shape":"LicenseSpecifications"},
        "RemoveLicenseSpecifications":{"shape":"LicenseSpecifications"}
      }
    },
    "UpdateLicenseSpecificationsForResourceResponse":{
      "type":"structure",
      "members":{
      }
    },
    "UpdateServiceSettingsRequest":{
      "type":"structure",
      "members":{
        "S3BucketArn":{"shape":"String"},
        "SnsTopicArn":{"shape":"String"},
        "OrganizationConfiguration":{"shape":"OrganizationConfiguration"},
        "EnableCrossAccountsDiscovery":{"shape":"BoxBoolean"}
      }
    },
    "UpdateServiceSettingsResponse":{
      "type":"structure",
      "members":{
      }
    },
    "UsageOperation":{
      "type":"string",
      "max":50
    },
    "ValidationException":{
      "type":"structure",
      "members":{
        "Message":{"shape":"Message"}
      },
      "exception":true
    }
  }
}
{
  "version":"2.0",
  "metadata":{
```

```
        "apiVersion":"2018-05-10",
        "endpointPrefix":"license-manager-linux-subscriptions",
        "jsonVersion":"1.1",
        "protocol":"rest-json",
        "serviceFullName":"AWS License Manager Linux Subscriptions",
        "serviceId":"License Manager Linux Subscriptions",
        "signatureVersion":"v4",
        "signingName":"license-manager-linux-subscriptions",
        "uid":"license-manager-linux-subscriptions-2018-05-10"
    },
    "operations":{
        "GetServiceSettings":{
            "name":"GetServiceSettings",
            "http":{
                "method":"POST",
                "requestUri":"/subscription/GetServiceSettings",
                "responseCode":200
            },
            "input":{"shape":"GetServiceSettingsRequest"},
            "output":{"shape":"GetServiceSettingsResponse"},
            "errors":[
                {"shape":"InternalServerException"},
                {"shape":"ThrottlingException"},
                {"shape":"ValidationException"}
            ],
            "idempotent":true
        },
        "ListLinuxSubscriptionInstances":{
            "name":"ListLinuxSubscriptionInstances",
            "http":{
                "method":"POST",
                "requestUri":"/subscription/ListLinuxSubscriptionInstances",
                "responseCode":200
            },
            "input":{"shape":"ListLinuxSubscriptionInstancesRequest"},
            "output":{"shape":"ListLinuxSubscriptionInstancesResponse"},
            "errors":[
                {"shape":"InternalServerException"},
                {"shape":"ThrottlingException"},
                {"shape":"ValidationException"}
            ],
            "idempotent":true
        },
        "ListLinuxSubscriptions":{
            "name":"ListLinuxSubscriptions",
            "http":{
                "method":"POST",
                "requestUri":"/subscription/ListLinuxSubscriptions",
```

```
      "responseCode":200
    },
    "input":{"shape":"ListLinuxSubscriptionsRequest"},
    "output":{"shape":"ListLinuxSubscriptionsResponse"},
    "errors":[
      {"shape":"InternalServerException"},
      {"shape":"ThrottlingException"},
      {"shape":"ValidationException"}
    ],
    "idempotent":true

  },
  "UpdateServiceSettings":{
    "name":"UpdateServiceSettings",
    "http":{
      "method":"POST",
      "requestUri":"/subscription/UpdateServiceSettings",
      "responseCode":200
    },
    "input":{"shape":"UpdateServiceSettingsRequest"},
    "output":{"shape":"UpdateServiceSettingsResponse"},
    "errors":[
      {"shape":"InternalServerException"},
      {"shape":"ThrottlingException"},
      {"shape":"ValidationException"}
    ],
    "idempotent":true
  }
},
"shapes":{
  "Boolean":{
    "type":"boolean",
    "box":true
  },
  "BoxInteger":{
    "type":"integer",
    "box":true
  },
  "BoxLong":{
    "type":"long",
    "box":true
  },
  "Filter":{
    "type":"structure",
    "members":{
      "Name":{"shape":"String"},
      "Operator":{"shape":"Operator"},
      "Values":{"shape":"StringList"}
```

```
    }
  },
  "FilterList":{
   "type":"list",
   "member":{"shape":"Filter"}
  },
  "GetServiceSettingsRequest":{
    "type":"structure",
   "members":{
   }
  },
  "GetServiceSettingsResponse":{
   "type":"structure",
   "members":{
    "HomeRegions":{"shape":"StringList"},
    "LinuxSubscriptionsDiscovery":{"shape":"LinuxSubscriptionsDiscovery"},
    "LinuxSubscriptionsDiscoverySettings":{"shape":"LinuxSubscriptionsDiscoverySettings"},
    "Status":{"shape":"Status"},
    "StatusMessage":{"shape":"StringMap"}
   }
  },
  "Instance":{
   "type":"structure",
   "members":{
    "AccountID":{"shape":"String"},
    "AmiId":{"shape":"String"},
    "InstanceID":{"shape":"String"},
    "InstanceType":{"shape":"String"},
    "LastUpdatedTime":{"shape":"String"},
    "ProductCode":{"shape":"ProductCodeList"},
    "Region":{"shape":"String"},
    "Status":{"shape":"String"},
    "SubscriptionName":{"shape":"String"},
    "UsageOperation":{"shape":"String"}
   }
  },
  "InstanceList":{

"type":"list",
   "member":{"shape":"Instance"}
  },
  "InternalServerException":{
   "type":"structure",
   "members":{
    "message":{"shape":"String"}
   },
   "exception":true,
   "fault":true
```

```
    },
    "LinuxSubscriptionsDiscovery":{
     "type":"string",
     "enum":[
       "Enabled",
       "Disabled"
     ]
    },
    "LinuxSubscriptionsDiscoverySettings":{
     "type":"structure",
     "required":[
       "OrganizationIntegration",
       "SourceRegions"
     ],
     "members":{
       "OrganizationIntegration":{"shape":"OrganizationIntegration"},
       "SourceRegions":{"shape":"StringList"}
     }
    },
    "ListLinuxSubscriptionInstancesRequest":{
     "type":"structure",
     "members":{
       "Filters":{"shape":"FilterList"},
       "MaxResults":{"shape":"BoxInteger"},
       "NextToken":{"shape":"ListLinuxSubscriptionInstancesRequestNextTokenString"}
     }
    },
    "ListLinuxSubscriptionInstancesRequestNextTokenString":{
     "type":"string",
     "max":16384,
     "min":1
    },
    "ListLinuxSubscriptionInstancesResponse":{
     "type":"structure",
     "members":{
       "Instances":{"shape":"InstanceList"},
       "NextToken":{"shape":"String"}
     }
    },
    "ListLinuxSubscriptionsRequest":{
     "type":"structure",
     "members":{
       "Filters":{"shape":"FilterList"},
       "MaxResults":{"shape":"BoxInteger"},
       "NextToken":{"shape":"ListLinuxSubscriptionsRequestNextTokenString"}
     }
    },
    "ListLinuxSubscriptionsRequestNextTokenString":{
```

```
   "type":"string",
   "max":16384,
   "min":1
  },
  "ListLinuxSubscriptionsResponse":{
   "type":"structure",
   "members":{
    "NextToken":{"shape":"String"},
    "Subscriptions":{"shape":"SubscriptionList"}
   }
  },
  "Operator":{
   "type":"string",
   "enum":[
    "Equal",
    "NotEqual",
    "Contains"
   ],
   "max":20,
   "min":1
  },
  "OrganizationIntegration":{
   "type":"string",
   "enum":[
    "Enabled",
    "Disabled"
   ]
  },
  "ProductCodeList":{
   "type":"list",
   "member":{"shape":"String"}
  },
  "Status":{
   "type":"string",
   "enum":[
    "InProgress",
    "Completed",
    "Successful",
    "Failed"
   ]
  },
  "String":{"type":"string"},
  "StringList":{
   "type":"list",
   "member":{"shape":"StringListMemberString"},
   "max":100,
   "min":1
  },
```

"StringListMemberString":{
 "type":"string",
 "max":100,
 "min":1
},
"StringMap":{
 "type":"map",
 "key":{"shape":"String"},
 "value":{"shape":"String"}
},
"Subscription":{
 "type":"structure",
 "members":{
  "InstanceCount":{"shape":"BoxLong"},
  "Name":{"shape":"String"},
  "Type":{"shape":"String"}
 }
},
"SubscriptionList":{
 "type":"list",
 "member":{"shape":"Subscription"}
},
"ThrottlingException":{
 "type":"structure",
 "members":{
  "message":{"shape":"String"}
 },
 "exception":true
},
"UpdateServiceSettingsRequest":{
 "type":"structure",
 "required":[
  "LinuxSubscriptionsDiscovery",
  "LinuxSubscriptionsDiscoverySettings"
 ],
 "members":{
  "AllowUpdate":{"shape":"Boolean"},
  "LinuxSubscriptionsDiscovery":{"shape":"LinuxSubscriptionsDiscovery"},
  "LinuxSubscriptionsDiscoverySettings":{"shape":"LinuxSubscriptionsDiscoverySettings"}
 }
},
"UpdateServiceSettingsResponse":{
 "type":"structure",
 "members":{
  "HomeRegions":{"shape":"StringList"},
  "LinuxSubscriptionsDiscovery":{"shape":"LinuxSubscriptionsDiscovery"},
  "LinuxSubscriptionsDiscoverySettings":{"shape":"LinuxSubscriptionsDiscoverySettings"},
  "Status":{"shape":"Status"},

```
        "StatusMessage":{"shape":"StringMap"}
      }
    },
    "ValidationException":{
      "type":"structure",
      "members":{
        "message":{"shape":"String"}
      },
      "exception":true
    }
  }
}
{
  "version": "2.0",
  "service": "<p>License Manager makes it easier to manage licenses from software vendors across multiple Amazon
Web Services accounts and on-premises servers.</p>",
  "operations": {
    "AcceptGrant": "<p>Accepts the specified grant.</p>",
    "CheckInLicense": "<p>Checks in the specified license. Check in a license when it is no longer in use.</p>",
    "CheckoutBorrowLicense": "<p>Checks out the specified license for offline use.</p>",
    "CheckoutLicense": "<p>Checks out the specified license.</p> <note> <p>If the account that created the license
is the same that is performing the check out, you must specify the account as the beneficiary.</p> </note>",
    "CreateGrant": "<p>Creates a grant for the specified license. A grant shares the use of license entitlements with a
specific Amazon Web Services account, an organization, or an organizational unit (OU). For more information, see
<a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted
licenses in License Manager</a> in the <i>License Manager User Guide</i>.</p>",
    "CreateGrantVersion": "<p>Creates a new version of the specified grant. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-licenses.html\">Granted licenses in
License Manager</a> in the <i>License Manager User Guide</i>.</p>",
    "CreateLicense": "<p>Creates a license.</p>",
    "CreateLicenseConfiguration": "<p>Creates a license configuration.</p> <p>A license configuration is an
abstraction of a customer license agreement that can be consumed and enforced by License Manager. Components
include specifications for the license type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared
tenancy, Dedicated Instance, Dedicated Host, or all of these), license affinity to host (how long a license must be
associated with a host), and the number of licenses purchased and used.</p>",
    "CreateLicenseConversionTaskForResource":
  "<p>Creates a new license conversion task.</p>",
    "CreateLicenseManagerReportGenerator": "<p>Creates a report generator.</p>",
    "CreateLicenseVersion": "<p>Creates a new version of the specified license.</p>",
    "CreateToken": "<p>Creates a long-lived token.</p> <p>A refresh token is a JWT token used to get an access
token. With an access token, you can call AssumeRoleWithWebIdentity to get role credentials that you can use to
call License Manager to manage the specified license.</p>",
    "DeleteGrant": "<p>Deletes the specified grant.</p>",
    "DeleteLicense": "<p>Deletes the specified license.</p>",
    "DeleteLicenseConfiguration": "<p>Deletes the specified license configuration.</p> <p>You cannot delete a
license configuration that is in use.</p>",
    "DeleteLicenseManagerReportGenerator": "<p>Deletes the specified report generator.</p> <p>This action deletes
the report generator, which stops it from generating future reports. The action cannot be reversed. It
```

has no effect on the previous reports from this generator.</p>",

"DeleteToken": "<p>Deletes the specified token. Must be called in the license home Region.</p>",

"ExtendLicenseConsumption": "<p>Extends the expiration date for license consumption.</p>",

"GetAccessToken": "<p>Gets a temporary access token to use with AssumeRoleWithWebIdentity. Access tokens are valid for one hour.</p>",

"GetGrant": "<p>Gets detailed information about the specified grant.</p>",

"GetLicense": "<p>Gets detailed information about the specified license.</p>",

"GetLicenseConfiguration": "<p>Gets detailed information about the specified license configuration.</p>",

"GetLicenseConversionTask": "<p>Gets information about the specified license type conversion task.</p>",

"GetLicenseManagerReportGenerator": "<p>Gets information about the specified report generator.</p>",

"GetLicenseUsage": "<p>Gets detailed information about the usage of the specified license.</p>",

"GetServiceSettings":
"<p>Gets the License Manager settings for the current Region.</p>",

"ListAssociationsForLicenseConfiguration": "<p>Lists the resource associations for the specified license configuration.</p> <p>Resource associations need not consume licenses from a license configuration. For example, an AMI or a stopped instance might not consume a license (depending on the license rules).</p>",

"ListDistributedGrants": "<p>Lists the grants distributed for the specified license.</p>",

"ListFailuresForLicenseConfigurationOperations": "<p>Lists the license configuration operations that failed.</p>",

"ListLicenseConfigurations": "<p>Lists the license configurations for your account.</p>",

"ListLicenseConversionTasks": "<p>Lists the license type conversion tasks for your account.</p>",

"ListLicenseManagerReportGenerators": "<p>Lists the report generators for your account.</p>",

"ListLicenseSpecificationsForResource": "<p>Describes the license configurations for the specified resource.</p>",

"ListLicenseVersions": "<p>Lists all versions of the specified license.</p>",

"ListLicenses": "<p>Lists the licenses for your account.</p>",

"ListReceivedGrants": "<p>Lists grants that are received. Received grants are grants created while specifying the recipient as this Amazon Web Services account, your organization, or an organizational unit (OU) to which this member account belongs.</p>",

"ListReceivedGrantsForOrganization": "<p>Lists the grants received for all accounts in the organization.</p>",

"ListReceivedLicenses": "<p>Lists received licenses.</p>",

"ListReceivedLicensesForOrganization": "<p>Lists the licenses received for all accounts in the organization.</p>",

"ListResourceInventory": "<p>Lists resources managed using Systems Manager inventory.</p>",

"ListTagsForResource": "<p>Lists the tags for the specified license configuration.</p>",

"ListTokens": "<p>Lists your tokens.</p>",

"ListUsageForLicenseConfiguration":
"<p>Lists all license usage records for a license configuration, displaying license consumption details by resource at a selected point in time. Use this action to audit the current license consumption for any license inventory and configuration.</p>",

"RejectGrant": "<p>Rejects the specified grant.</p>",

"TagResource": "<p>Adds the specified tags to the specified license configuration.</p>",

"UntagResource": "<p>Removes the specified tags from the specified license configuration.</p>",

"UpdateLicenseConfiguration": "<p>Modifies the attributes of an existing license configuration.</p>",

"UpdateLicenseManagerReportGenerator": "<p>Updates a report generator.</p> <p>After you make changes to a report generator, it starts generating new reports within 60 minutes of being updated.</p>",

"UpdateLicenseSpecificationsForResource": "<p>Adds or removes the specified license configurations for the specified Amazon Web Services resource.</p> <p>You can update the

license specifications of AMIs, instances, and hosts. You cannot update the license specifications for launch templates and CloudFormation templates, as they send license configurations to the operation that creates the resource.</p>",
  "UpdateServiceSettings": "<p>Updates License Manager settings for the current Region.</p>"
  },
 "shapes": {
  "AcceptGrantRequest": {
  "base": null,
  "refs": {
  }
  },
  "AcceptGrantResponse": {
  "base": null,
  "refs": {
  }
  },
  "AccessDeniedException": {
  "base": "<p>Access to resource denied.</p>",
  "refs": {
  }
  },
  "ActivationOverrideBehavior": {
  "base": null,
  "refs": {
   "Options$ActivationOverrideBehavior": "<p>An activation option for your grant that determines the behavior of activating a grant. Activation options can only be used with granted licenses sourced from the Amazon Web Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the <code>Status</code> parameter.</p>   <p>As a license administrator, you can optionally specify an <code>ActivationOverrideBehavior</code> when activating a grant.</p>   <p>As a grantor, you can optionally specify an <code>ActivationOverrideBehavior</code> when you activate a grant for a grantee account in your organization.</p>   <p>As a grantee, if the grantor creating the distributed grant doesnt specify an <code>ActivationOverrideBehavior</code>, you can optionally specify one when you are activating the grant.</p>   <dl> <dt>DISTRIBUTED_GRANTS_ONLY</dt> <dd> <p>Use this value to activate a grant without replacing any member accounts active grants for the same product.</p> </dd> <dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt> <dd> <p>Use this value to activate a grant and disable other active grants in any member accounts for the same product. This action will also replace their previously activated grants with this activated grant.</p> </dd> </dl>"
  }
  },
  "AllowedOperation": {
  "base": null,
  "refs": {
   "AllowedOperationList$member": null
  }
  },
  "AllowedOperationList": {
  "base": null,

```
    "refs": {
      "CreateGrantRequest$AllowedOperations": "<p>Allowed operations for the grant.</p>",
      "CreateGrantVersionRequest$AllowedOperations": "<p>Allowed operations for the grant.</p>",
      "Grant$GrantedOperations": "<p>Granted operations.</p>",
      "ReceivedMetadata$AllowedOperations": "<p>Allowed operations.</p>"
    }
  },
  "Arn": {
    "base": null,
    "refs": {
      "AcceptGrantRequest$GrantArn": "<p>Amazon Resource Name (ARN) of the grant.</p>",
      "AcceptGrantResponse$GrantArn": "<p>Grant ARN.</p>",
      "ArnList$member": null,
      "CheckoutBorrowLicenseRequest$LicenseArn": "<p>Amazon Resource Name (ARN) of the license. The
license must use the borrow consumption configuration.</p>",
      "CheckoutBorrowLicenseResponse$LicenseArn":
 "<p>Amazon Resource Name (ARN) of the license.</p>",
      "CreateGrantRequest$LicenseArn": "<p>Amazon Resource Name (ARN) of the license.</p>",
      "CreateGrantResponse$GrantArn": "<p>Grant ARN.</p>",
      "CreateGrantVersionRequest$GrantArn": "<p>Amazon Resource Name (ARN) of the grant.</p>",
      "CreateGrantVersionResponse$GrantArn": "<p>Grant ARN.</p>",
      "CreateLicenseConversionTaskForResourceRequest$ResourceArn": "<p>Amazon Resource Name (ARN) of
the resource you are converting the license type for.</p>",
      "CreateLicenseResponse$LicenseArn": "<p>Amazon Resource Name (ARN) of the license.</p>",
      "CreateLicenseVersionRequest$LicenseArn": "<p>Amazon Resource Name (ARN) of the license.</p>",
      "CreateLicenseVersionResponse$LicenseArn": "<p>License ARN.</p>",
      "CreateTokenRequest$LicenseArn": "<p>Amazon Resource Name (ARN) of the license. The ARN is mapped
to the aud claim of the JWT token.</p>",
      "DeleteGrantRequest$GrantArn":
 "<p>Amazon Resource Name (ARN) of the grant.</p>",
      "DeleteGrantResponse$GrantArn": "<p>Grant ARN.</p>",
      "DeleteLicenseRequest$LicenseArn": "<p>Amazon Resource Name (ARN) of the license.</p>",
      "GetGrantRequest$GrantArn": "<p>Amazon Resource Name (ARN) of the grant.</p>",
      "GetLicenseRequest$LicenseArn": "<p>Amazon Resource Name (ARN) of the license.</p>",
      "GetLicenseUsageRequest$LicenseArn": "<p>Amazon Resource Name (ARN) of the license.</p>",
      "Grant$GrantArn": "<p>Amazon Resource Name (ARN) of the grant.</p>",
      "Grant$ParentArn": "<p>Parent ARN.</p>",
      "Grant$LicenseArn": "<p>License ARN.</p>",
      "Grant$GranteePrincipalArn": "<p>The grantee principal ARN.</p>",
      "GrantedLicense$LicenseArn": "<p>Amazon Resource Name (ARN) of the license.</p>",
      "License$LicenseArn": "<p>Amazon Resource Name (ARN) of the license.</p>",
      "ListLicenseVersionsRequest$LicenseArn": "<p>Amazon Resource Name (ARN) of
 the license.</p>",
      "ListReceivedGrantsForOrganizationRequest$LicenseArn": "<p>The Amazon Resource Name (ARN) of the
received license.</p>",
      "PrincipalArnList$member": null,
      "RejectGrantRequest$GrantArn": "<p>Amazon Resource Name (ARN) of the grant.</p>",
      "RejectGrantResponse$GrantArn": "<p>Grant ARN.</p>"
```

```
      }
    },
    "ArnList": {
      "base": null,
      "refs": {
        "CreateTokenRequest$RoleArns": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the
token. License Manager does not check whether the roles are in use.</p>",
        "ListDistributedGrantsRequest$GrantArns": "<p>Amazon Resource Names (ARNs) of the grants.</p>",
        "ListLicensesRequest$LicenseArns": "<p>Amazon Resource Names (ARNs) of the licenses.</p>",
        "ListReceivedGrantsRequest$GrantArns": "<p>Amazon Resource Names (ARNs) of the grants.</p>",
        "ListReceivedLicensesRequest$LicenseArns": "<p>Amazon Resource Names (ARNs) of the licenses.</p>",

        "ReportContext$licenseConfigurationArns": "<p>Amazon Resource Name (ARN) of the license configuration
that this generator reports on.</p>",
        "TokenData$RoleArns": "<p>Amazon Resource Names (ARN) of the roles included in the token.</p>"
      }
    },
    "AuthorizationException": {
      "base": "<p>The Amazon Web Services user account does not have permission to perform the action. Check the
IAM policy associated with this account.</p>",
      "refs": {
      }
    },
    "AutomatedDiscoveryInformation": {
      "base": "<p>Describes automated discovery.</p>",
      "refs": {
        "GetLicenseConfigurationResponse$AutomatedDiscoveryInformation": "<p>Automated discovery
information.</p>",
        "LicenseConfiguration$AutomatedDiscoveryInformation": "<p>Automated discovery information.</p>"
      }
    },
    "Boolean": {
      "base": null,
      "refs": {
        "ExtendLicenseConsumptionRequest$DryRun": "<p>Checks whether you have the required permissions for the
action,
without actually making the request. Provides an error response if you do not have the required permissions.</p>",
        "OrganizationConfiguration$EnableIntegration": "<p>Enables Organizations integration.</p>"
      }
    },
    "BorrowConfiguration": {
      "base": "<p>Details about a borrow configuration.</p>",
      "refs": {
        "ConsumptionConfiguration$BorrowConfiguration": "<p>Details about a borrow configuration.</p>"
      }
    },
    "BoxBoolean": {
      "base": null,
```

    "refs": {
      "BorrowConfiguration$AllowEarlyCheckIn": "<p>Indicates whether early check-ins are allowed.</p>",
      "CreateLicenseConfigurationRequest$LicenseCountHardLimit": "<p>Indicates whether hard or soft license enforcement is used. Exceeding a hard limit blocks the launch of new instances.</p>",
      "CreateLicenseConfigurationRequest$DisassociateWhenNotFound": "<p>When true, disassociates a resource when software is uninstalled.</p>",
      "Entitlement$Overage": "<p>Indicates
 whether overages are allowed.</p>",
      "Entitlement$AllowCheckIn": "<p>Indicates whether check-ins are allowed.</p>",
      "GetLicenseConfigurationResponse$LicenseCountHardLimit": "<p>Sets the number of available licenses as a hard limit.</p>",
      "GetLicenseConfigurationResponse$DisassociateWhenNotFound": "<p>When true, disassociates a resource when software is uninstalled.</p>",
      "GetServiceSettingsResponse$EnableCrossAccountsDiscovery": "<p>Indicates whether cross-account discovery is enabled.</p>",
      "LicenseConfiguration$LicenseCountHardLimit": "<p>Number of available licenses as a hard limit.</p>",
      "LicenseConfiguration$DisassociateWhenNotFound": "<p>When true, disassociates a resource when software is uninstalled.</p>",
      "UpdateLicenseConfigurationRequest$LicenseCountHardLimit": "<p>New hard limit of the number of available licenses.</p>",
      "UpdateLicenseConfigurationRequest$DisassociateWhenNotFound": "<p>When true, disassociates
 a resource when software is uninstalled.</p>",
      "UpdateServiceSettingsRequest$EnableCrossAccountsDiscovery": "<p>Activates cross-account discovery.</p>"
    }
  },
  "BoxInteger": {
    "base": null,
    "refs": {
      "BorrowConfiguration$MaxTimeToLiveInMinutes": "<p>Maximum time for the borrow configuration, in minutes.</p>",
      "ListAssociationsForLicenseConfigurationRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListFailuresForLicenseConfigurationOperationsRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListLicenseConfigurationsRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListLicenseConversionTasksRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListLicenseSpecificationsForResourceRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",

    "ListResourceInventoryRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListUsageForLicenseConfigurationRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ProvisionalConfiguration$MaxTimeToLiveInMinutes": "<p>Maximum time for the provisional configuration, in minutes.</p>"
    }

```
    },
  "BoxLong": {
   "base": null,
   "refs": {
     "ConsumedLicenseSummary$ConsumedLicenses": "<p>Number of licenses consumed by the resource.</p>",
     "CreateLicenseConfigurationRequest$LicenseCount": "<p>Number of licenses managed by the license
configuration.</p>",
     "GetLicenseConfigurationResponse$LicenseCount": "<p>Number of available licenses.</p>",
     "GetLicenseConfigurationResponse$ConsumedLicenses": "<p>Number of licenses assigned to resources.</p>",
     "LicenseConfiguration$LicenseCount": "<p>Number of licenses managed by the license configuration.</p>",
     "LicenseConfiguration$ConsumedLicenses":
 "<p>Number of licenses consumed. </p>",
     "LicenseConfigurationUsage$ConsumedLicenses": "<p>Number of licenses consumed by the resource.</p>",
     "ManagedResourceSummary$AssociationCount": "<p>Number of resources associated with licenses.</p>",
     "UpdateLicenseConfigurationRequest$LicenseCount": "<p>New number of licenses managed by the license
configuration.</p>"
   }
  },
  "CheckInLicenseRequest": {
   "base": null,
   "refs": {
   }
  },
  "CheckInLicenseResponse": {
   "base": null,
   "refs": {
   }
  },
  "CheckoutBorrowLicenseRequest": {
   "base": null,
   "refs": {
   }
  },
  "CheckoutBorrowLicenseResponse": {
   "base": null,
   "refs": {
   }
  },
  "CheckoutLicenseRequest": {
   "base": null,
   "refs": {
   }
  },
  "CheckoutLicenseResponse": {
   "base": null,
   "refs": {
   }
  },
```

```
   "CheckoutType": {
    "base": null,
    "refs": {
      "CheckoutLicenseRequest$CheckoutType": "<p>Checkout type.</p>",
      "CheckoutLicenseResponse$CheckoutType": "<p>Checkout type.</p>"
    }
   },
   "ClientRequestToken": {
    "base": null,
    "refs": {
      "CreateLicenseManagerReportGeneratorRequest$ClientToken": "<p>Unique, case-sensitive identifier that you
provide to ensure the idempotency of the request.</p>",
      "UpdateLicenseManagerReportGeneratorRequest$ClientToken": "<p>Unique, case-sensitive identifier that you
provide to ensure the idempotency of the request.</p>"
    }
   },
   "ClientToken": {
    "base": null,
    "refs": {
      "CheckoutBorrowLicenseRequest$ClientToken": "<p>Unique, case-sensitive identifier that you provide to
ensure the idempotency of the request.</p>",
      "CheckoutLicenseRequest$ClientToken": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
      "CreateGrantRequest$ClientToken": "<p>Unique, case-sensitive
 identifier that you provide to ensure the idempotency of the request.</p>",
      "CreateGrantVersionRequest$ClientToken": "<p>Unique, case-sensitive identifier that you provide to ensure
the idempotency of the request.</p>",
      "CreateLicenseRequest$ClientToken": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
      "CreateLicenseVersionRequest$ClientToken": "<p>Unique, case-sensitive identifier that you provide to ensure
the idempotency of the request.</p>",
      "CreateTokenRequest$ClientToken": "<p>Idempotency token, valid for 10 minutes.</p>"
    }
   },
   "ConflictException": {
    "base": "<p>There was a conflict processing the request. Try your request again.</p>",
    "refs": {
    }
   },
   "ConsumedLicenseSummary": {
    "base": "<p>Details about license consumption.</p>",
    "refs": {
      "ConsumedLicenseSummaryList$member": null
    }
   },
   "ConsumedLicenseSummaryList": {
    "base": null,
    "refs": {
```

```
    "GetLicenseConfigurationResponse$ConsumedLicenseSummaryList": "<p>Summaries of the licenses
consumed by resources.</p>",
      "LicenseConfiguration$ConsumedLicenseSummaryList": "<p>Summaries for licenses consumed by various
resources.</p>"
    }
  },
  "ConsumptionConfiguration": {
    "base": "<p>Details about a consumption configuration.</p>",
    "refs": {
      "CreateLicenseRequest$ConsumptionConfiguration": "<p>Configuration for consumption of the license.
Choose a provisional configuration for workloads running with continuous connectivity. Choose a borrow
configuration for workloads with offline usage.</p>",
      "CreateLicenseVersionRequest$ConsumptionConfiguration": "<p>Configuration for consumption of the
license. Choose a provisional configuration for workloads running with continuous connectivity. Choose a borrow
configuration for workloads with offline usage.</p>",
      "GrantedLicense$ConsumptionConfiguration":
 "<p>Configuration for consumption of the license.</p>",
      "License$ConsumptionConfiguration": "<p>Configuration for consumption of the license.</p>"
    }
  },
  "CreateGrantRequest": {
    "base": null,
    "refs": {
    }
  },
  "CreateGrantResponse": {
    "base": null,
    "refs": {
    }
  },
  "CreateGrantVersionRequest": {
    "base": null,
    "refs": {
    }
  },
  "CreateGrantVersionResponse": {
    "base": null,
    "refs": {
    }
  },
  "CreateLicenseConfigurationRequest": {
    "base": null,
    "refs": {
    }
  },
  "CreateLicenseConfigurationResponse": {
    "base": null,
    "refs": {
```

```
    }
  },
  "CreateLicenseConversionTaskForResourceRequest": {
   "base": null,
   "refs": {
   }
  },
  "CreateLicenseConversionTaskForResourceResponse": {
   "base": null,
   "refs": {
   }
  },
  "CreateLicenseManagerReportGeneratorRequest": {
   "base": null,
   "refs": {

   }
  },
  "CreateLicenseManagerReportGeneratorResponse": {
   "base": null,
   "refs": {
   }
  },
  "CreateLicenseRequest": {
   "base": null,
   "refs": {
   }
  },
  "CreateLicenseResponse": {
   "base": null,
   "refs": {
   }
  },
  "CreateLicenseVersionRequest": {
   "base": null,
   "refs": {
   }
  },
  "CreateLicenseVersionResponse": {
   "base": null,
   "refs": {
   }
  },
  "CreateTokenRequest": {
   "base": null,
   "refs": {
   }
  },
```

"CreateTokenResponse": {
 "base": null,
 "refs": {
 }
},
"DateTime": {
 "base": null,
 "refs": {
  "AutomatedDiscoveryInformation$LastRunTime": "<p>Time that automated discovery last ran.</p>",
  "GetLicenseConversionTaskResponse$StartTime": "<p>Time at which the license type conversion task was started .</p>",
  "GetLicenseConversionTaskResponse$LicenseConversionTime": "<p>Amount of time to complete the license type conversion.</p>",
  "GetLicenseConversionTaskResponse$EndTime": "<p>Time at which the license type conversion task was completed.</p>",
  "LicenseConfigurationAssociation$AssociationTime": "<p>Time when the license configuration was associated with the resource.</p>",
  "LicenseConfigurationUsage$AssociationTime": "<p>Time when the license configuration was initially associated with the resource.</p>",
  "LicenseConversionTask$StartTime": "<p>The time the conversion task was started at.</p>",
  "LicenseConversionTask$LicenseConversionTime": "<p>The time the usage operation value of the resource was changed.</p>",
  "LicenseConversionTask$EndTime": "<p>The time the conversion task was completed.</p>",
  "LicenseOperationFailure$FailureTime": "<p>Failure time.</p>"
 }
},
"DatetimeRange": {
 "base": "<p>Describes a time range, in ISO8601-UTC format.</p>",
 "refs": {
  "CreateLicenseRequest$Validity": "<p>Date and time range during which the license is valid, in ISO8601-UTC format.</p>",
  "CreateLicenseVersionRequest$Validity": "<p>Date and time range during which the license is valid, in ISO8601-UTC format.</p>",
  "GrantedLicense$Validity": "<p>Date and time range during which the granted license is valid, in ISO8601-UTC format.</p>",
  "License$Validity": "<p>Date and time range during which the license is valid, in ISO8601-UTC format.</p>"
 }
},
"DeleteGrantRequest": {
 "base": null,
 "refs": {
 }
},
"DeleteGrantResponse": {
 "base": null,
 "refs": {
 }
},

```
"DeleteLicenseConfigurationRequest": {
 "base": null,
 "refs": {
 }
},
"DeleteLicenseConfigurationResponse": {
 "base": null,
 "refs": {
 }
},
"DeleteLicenseManagerReportGeneratorRequest": {
 "base": null,
 "refs": {
 }
},
"DeleteLicenseManagerReportGeneratorResponse": {
 "base": null,
 "refs": {
 }
},
"DeleteLicenseRequest": {
 "base": null,
 "refs": {
 }
},
"DeleteLicenseResponse": {
 "base": null,
 "refs": {
 }
},
"DeleteTokenRequest": {
 "base": null,
 "refs": {
 }
},
"DeleteTokenResponse": {
 "base": null,
 "refs": {
 }
},
"DigitalSignatureMethod": {
 "base": null,
 "refs": {
   "CheckoutBorrowLicenseRequest$DigitalSignatureMethod": "<p>Digital signature method. The possible value
is JSON Web Signature (JWS) algorithm PS384. For more information, see <a
href=\"https://tools.ietf.org/html/rfc7518#section-3.5\">RFC 7518 Digital Signature with RSASSA-PSS</a>.</p>"
 }
},
```

```
    "Entitlement": {
      "base": "<p>Describes a resource entitled for use with a license.</p>",
      "refs": {
        "EntitlementList$member": null
      }
    },
    "EntitlementData": {
      "base": "<p>Data associated with an entitlement resource.</p>",

      "refs": {
        "EntitlementDataList$member": null
      }
    },
    "EntitlementDataList": {
      "base": null,
      "refs": {
        "CheckoutBorrowLicenseRequest$Entitlements": "<p>License entitlements. Partial checkouts are not
supported.</p>",
        "CheckoutBorrowLicenseResponse$EntitlementsAllowed": "<p>Allowed license entitlements.</p>",
        "CheckoutLicenseRequest$Entitlements": "<p>License entitlements.</p>",
        "CheckoutLicenseResponse$EntitlementsAllowed": "<p>Allowed license entitlements.</p>"
      }
    },
    "EntitlementDataUnit": {
      "base": null,
      "refs": {
        "EntitlementData$Unit": "<p>Entitlement data unit.</p>",
        "EntitlementUsage$Unit": "<p>Entitlement usage unit.</p>"
      }
    },
    "EntitlementList": {
      "base": null,
      "refs": {
        "CreateLicenseRequest$Entitlements": "<p>License entitlements.</p>",
        "CreateLicenseVersionRequest$Entitlements": "<p>License entitlements.</p>",
        "GrantedLicense$Entitlements":
"<p>License entitlements.</p>",
        "License$Entitlements": "<p>License entitlements.</p>"
      }
    },
    "EntitlementNotAllowedException": {
      "base": "<p>The entitlement is not allowed.</p>",
      "refs": {
      }
    },
    "EntitlementUnit": {
      "base": null,
      "refs": {
```

```
        "Entitlement$Unit": "<p>Entitlement unit.</p>"
      }
    },
    "EntitlementUsage": {
      "base": "<p>Usage associated with an entitlement resource.</p>",
      "refs": {
        "EntitlementUsageList$member": null
      }
    },
    "EntitlementUsageList": {
      "base": null,
      "refs": {
        "LicenseUsage$EntitlementUsages": "<p>License entitlement usages.</p>"
      }
    },
    "ExtendLicenseConsumptionRequest": {
      "base": null,
      "refs": {
      }
    },
    "ExtendLicenseConsumptionResponse": {
      "base": null,
      "refs": {
      }
    },
    "FailedDependencyException": {
      "base": "<p>A dependency required to run the API is missing.</p>",
      "refs": {
      }
    },
    "Filter": {
      "base": "<p>A filter name and value pair that is used to return more specific results from a describe operation.
Filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>",
      "refs": {
        "FilterList$member": null,
        "Filters$member": null
      }
    },
    "FilterLimitExceededException": {
      "base": "<p>The request uses too many filters or too many filter values.</p>",
      "refs": {
      }
    },
    "FilterList": {
      "base": null,
      "refs": {
        "ListDistributedGrantsRequest$Filters": "<p>Filters to scope the results. The following filters are
supported:</p> <ul> <li> <p> <code>LicenseArn</code> </p> </li> <li> <p> <code>GrantStatus</code> </p>
```

  <p> <code>GranteePrincipalARN</code> </p>   <p> <code>ProductSKU</code> </p>   <p> <code>LicenseIssuerName</code> </p> ",
    "ListLicenseManagerReportGeneratorsRequest$Filters":
 "<p>Filters to scope the results. The following filters are supported: </p>   <p> <code>LicenseConfigurationArn</code> </p> ",
    "ListLicensesRequest$Filters": "<p>Filters to scope the results. The following filters are supported:</p>   <p> <code>Beneficiary</code> </p>   <p> <code>ProductSKU</code> </p>   <p> <code>Fingerprint</code> </p>   <p> <code>Status</code> </p> ",
    "ListReceivedGrantsForOrganizationRequest$Filters": "<p>Filters to scope the results. The following filters are supported:</p>   <p> <code>ParentArn</code> </p>   <p> <code>GranteePrincipalArn</code> </p> ",
    "ListReceivedGrantsRequest$Filters": "<p>Filters to scope the results. The following filters are supported:</p>   <p> <code>ProductSKU</code> </p>   <p> <code>LicenseIssuerName</code> </p>   <p> <code>LicenseArn</code> </p>   <p> <code>GrantStatus</code>
 </p>   <p> <code>GranterAccountId</code> </p> ",
    "ListReceivedLicensesForOrganizationRequest$Filters": "<p>Filters to scope the results. The following filters are supported:</p>   <p> <code>Beneficiary</code> </p>   <p> <code>ProductSKU</code> </p> ",
    "ListReceivedLicensesRequest$Filters": "<p>Filters to scope the results. The following filters are supported:</p>   <p> <code>ProductSKU</code> </p>   <p> <code>Status</code> </p>   <p> <code>Fingerprint</code> </p>   <p> <code>IssuerName</code> </p>   <p> <code>Beneficiary</code> </p> ",
    "ListTokensRequest$Filters": "<p>Filters to scope the results. The following filter is supported:</p>   <p> <code>LicenseArns</code> </p> "
    }
  },
  "FilterName": {
   "base": null,
   "refs": {
    "Filter$Name": "<p>Name of the filter. Filter names are case-sensitive.</p>"
    }
  },
  "FilterValue": {
   "base": null,
   "refs": {
    "FilterValues$member": null
    }
  },
  "FilterValues": {
   "base": null,
   "refs": {
    "Filter$Values": "<p>The value of the filter, which is case-sensitive. You can only specify one value for the filter.</p>"
    }
  },
  "Filters": {
   "base": null,
   "refs": {

"ListLicenseConfigurationsRequest$Filters": "<p>Filters to scope the results. The following filters and logical operators are supported:</p>   <p> <code>licenseCountingType</code> - The dimension for which licenses are counted. Possible values are <code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>. Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>   <p> <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license enforcement is used. Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>   <p> <code>usagelimitExceeded</code> - A Boolean value that indicates whether the available licenses have been exceeded. Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p> ",
      "ListLicenseConversionTasksRequest$Filters": "<p> Filters to scope the results. Valid filters are <code>ResourceArns</code> and <code>Status</code>. </p>",
      "ListUsageForLicenseConfigurationRequest$Filters": "<p>Filters to scope the results. The following filters and logical operators are supported:</p>   <p> <code>resourceArn</code> - The ARN of the license configuration resource. Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>   <p> <code>resourceType</code> - The resource type (<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> | <code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>   <p> <code>resourceAccount</code> - The ID of the account that owns the resource. Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p> "
    }
  },
  "GetAccessTokenRequest": {
   "base": null,
   "refs": {
   }
  },
  "GetAccessTokenResponse": {
   "base": null,
   "refs": {
   }
  },
  "GetGrantRequest": {
   "base": null,
   "refs": {
   }
  },
  "GetGrantResponse": {
   "base": null,
   "refs": {
   }
  },
  "GetLicenseConfigurationRequest": {
   "base": null,
   "refs": {
   }
  },
  "GetLicenseConfigurationResponse": {

```
    "base": null,
    "refs": {
    }
  },
  "GetLicenseConversionTaskRequest": {
    "base": null,
    "refs": {
    }
  },
  "GetLicenseConversionTaskResponse": {
    "base": null,
    "refs": {
    }
  },
  "GetLicenseManagerReportGeneratorRequest": {
    "base": null,
    "refs": {
    }
  },
  "GetLicenseManagerReportGeneratorResponse": {
    "base":
null,
    "refs": {
    }
  },
  "GetLicenseRequest": {
    "base": null,
    "refs": {
    }
  },
  "GetLicenseResponse": {
    "base": null,
    "refs": {
    }
  },
  "GetLicenseUsageRequest": {
    "base": null,
    "refs": {
    }
  },
  "GetLicenseUsageResponse": {
    "base": null,
    "refs": {
    }
  },
  "GetServiceSettingsRequest": {
    "base": null,
    "refs": {
```

```
      }
    },
  "GetServiceSettingsResponse": {
   "base": null,
   "refs": {
   }
  },
  "Grant": {
   "base": "<p>Describes a grant.</p>",
   "refs": {
     "GetGrantResponse$Grant": "<p>Grant details.</p>",
     "GrantList$member": null
   }
  },
  "GrantList": {
   "base": null,
   "refs": {
     "ListDistributedGrantsResponse$Grants": "<p>Distributed grant details.</p>",
     "ListReceivedGrantsForOrganizationResponse$Grants": "<p>Lists the grants the organization has
received.</p>",
     "ListReceivedGrantsResponse$Grants":
 "<p>Received grant details.</p>"
   }
  },
  "GrantStatus": {
   "base": null,
   "refs": {
     "AcceptGrantResponse$Status": "<p>Grant status.</p>",
     "CreateGrantResponse$Status": "<p>Grant status.</p>",
     "CreateGrantVersionRequest$Status": "<p>Grant status.</p>",
     "CreateGrantVersionResponse$Status": "<p>Grant status.</p>",
     "DeleteGrantResponse$Status": "<p>Grant status.</p>",
     "Grant$GrantStatus": "<p>Grant status.</p>",
     "RejectGrantResponse$Status": "<p>Grant status.</p>"
   }
  },
  "GrantedLicense": {
   "base": "<p>Describes a license that is granted to a grantee.</p>",
   "refs": {
     "GrantedLicenseList$member": null
   }
  },
  "GrantedLicenseList": {
   "base": null,
   "refs": {
     "ListReceivedLicensesForOrganizationResponse$Licenses": "<p>Lists the licenses the organization has
received.</p>",
     "ListReceivedLicensesResponse$Licenses": "<p>Received license
```

```
        details.</p>"
      }
    },
    "ISO8601DateTime": {
      "base": null,
      "refs": {
        "CheckoutBorrowLicenseResponse$IssuedAt": "<p>Date and time at which the license checkout is
issued.</p>",
        "CheckoutBorrowLicenseResponse$Expiration": "<p>Date and time at which the license checkout
expires.</p>",
        "CheckoutLicenseResponse$IssuedAt": "<p>Date and time at which the license checkout is issued.</p>",
        "CheckoutLicenseResponse$Expiration": "<p>Date and time at which the license checkout expires.</p>",
        "DatetimeRange$Begin": "<p>Start of the time range.</p>",
        "DatetimeRange$End": "<p>End of the time range.</p>",
        "DeleteLicenseResponse$DeletionDate": "<p>Date when the license is deleted.</p>",
        "ExtendLicenseConsumptionResponse$Expiration": "<p>Date and time at which the license consumption
expires.</p>",
        "GrantedLicense$CreateTime": "<p>Creation time of the granted license.</p>",
        "License$CreateTime": "<p>License
 creation time.</p>",
        "TokenData$ExpirationTime": "<p>Token expiration time, in ISO8601-UTC format.</p>"
      }
    },
    "Integer": {
      "base": null,
      "refs": {
        "CreateTokenRequest$ExpirationInDays": "<p>Token expiration, in days, counted from token creation. The
default is 365 days.</p>",
        "ReportFrequency$value": "<p>Number of times within the frequency period that a report is generated. The
only supported value is <code>1</code>.</p>"
      }
    },
    "InvalidParameterValueException": {
      "base": "<p>One or more parameter values are not valid.</p>",
      "refs": {
      }
    },
    "InvalidResourceStateException": {
      "base": "<p>License Manager cannot allocate a license to a resource because of its state. </p> <p>For example,
you cannot allocate a license to an instance in the process of shutting down.</p>",
      "refs": {
      }
    },
    "InventoryFilter": {
      "base": "<p>An inventory filter.</p>",
      "refs": {
```

```
      "InventoryFilterList$member": null
    }
  },
  "InventoryFilterCondition": {
    "base": null,
    "refs": {
      "InventoryFilter$Condition": "<p>Condition of the filter.</p>"
    }
  },
  "InventoryFilterList": {
    "base": null,
    "refs": {
      "ListResourceInventoryRequest$Filters": "<p>Filters to scope the results. The following filters and logical
operators are supported:</p> <ul> <li> <p> <code>account_id</code> - The ID of the Amazon Web Services
account that owns the resource. Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p> </li> <li> <p> <code>application_name</code> - The name of the
application. Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p> </li> <li> <p>
<code>license_included</code> - The type of license included. Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>. Possible values are <code>sql-server-enterprise</code> | <code>sql-server-
standard</code> | <code>sql-server-web</code>
| <code>windows-server-datacenter</code>.</p> </li> <li> <p> <code>platform</code> - The platform of the
resource. Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p> </li> <li> <p>
<code>resource_id</code> - The ID of the resource. Logical operators are <code>EQUALS</code> |
<code>NOT_EQUALS</code>.</p> </li> <li> <p> <code>tag:&lt;key&gt;</code> - The key/value combination of
a tag assigned to the resource. Logical operators are <code>EQUALS</code> (single account) or
<code>EQUALS</code> | <code>NOT_EQUALS</code> (cross account).</p> </li> </ul>"
    }
  },
  "Issuer": {
    "base": "<p>Details about the issuer of a license.</p>",
    "refs": {
      "CreateLicenseRequest$Issuer": "<p>License issuer.</p>",
      "CreateLicenseVersionRequest$Issuer": "<p>License issuer.</p>"
    }
  },
  "IssuerDetails": {
    "base": "<p>Details associated with the issuer of a license.</p>",
    "refs": {
      "GrantedLicense$Issuer": "<p>Granted
license issuer.</p>",
      "License$Issuer": "<p>License issuer.</p>"
    }
  },
  "License": {
    "base": "<p>Software license that is managed in License Manager.</p>",
    "refs": {
      "GetLicenseResponse$License": "<p>License details.</p>",
      "LicenseList$member": null
```

```
      }
    },
    "LicenseConfiguration": {
      "base": "<p>A license configuration is an abstraction of a customer license agreement that can be consumed and
enforced by License Manager. Components include specifications for the license type (licensing by instance, socket,
CPU, or vCPU), allowed tenancy (shared tenancy, Dedicated Instance, Dedicated Host, or all of these), host affinity
(how long a VM must be associated with a host), and the number of licenses purchased and used.</p>",
      "refs": {
        "LicenseConfigurations$member": null
      }
    },
    "LicenseConfigurationAssociation": {
      "base": "<p>Describes an association with a license configuration.</p>",
      "refs": {
        "LicenseConfigurationAssociations$member": null
      }
    },
    "LicenseConfigurationAssociations": {
      "base": null,
      "refs": {
        "ListAssociationsForLicenseConfigurationResponse$LicenseConfigurationAssociations": "<p>Information
about the associations for the license configuration.</p>"
      }
    },
    "LicenseConfigurationStatus": {
      "base": null,
      "refs": {
        "UpdateLicenseConfigurationRequest$LicenseConfigurationStatus": "<p>New status of the license
configuration.</p>"
      }
    },
    "LicenseConfigurationUsage": {
      "base": "<p>Details about the usage of a resource associated with a license configuration.</p>",
      "refs": {
        "LicenseConfigurationUsageList$member": null
      }
    },
    "LicenseConfigurationUsageList": {
      "base": null,
      "refs": {
        "ListUsageForLicenseConfigurationResponse$LicenseConfigurationUsageList": "<p>Information about the
license configurations.</p>"
      }

    },
    "LicenseConfigurations": {
      "base": null,
      "refs": {
```

```
    "ListLicenseConfigurationsResponse$LicenseConfigurations": "<p>Information about the license
configurations.</p>"
    }
  },
  "LicenseConversionContext": {
    "base": "<p>Information about a license type conversion task.</p>",
    "refs": {
      "CreateLicenseConversionTaskForResourceRequest$SourceLicenseContext": "<p>Information that identifies
the license type you are converting from. For the structure of the source license, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/conversion-procedures.html#conversion-
cli\">Convert a license type using the CLI </a> in the <i>License Manager User Guide</i>.</p>",
      "CreateLicenseConversionTaskForResourceRequest$DestinationLicenseContext": "<p>Information that
identifies the license type you are converting to. For the structure of the destination license, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/conversion-procedures.html#conversion-
cli\">Convert
 a license type using the CLI </a> in the <i>License Manager User Guide</i>.</p>",
      "GetLicenseConversionTaskResponse$SourceLicenseContext": "<p>Information about the license type
converted from.</p>",
      "GetLicenseConversionTaskResponse$DestinationLicenseContext": "<p>Information about the license type
converted to.</p>",
      "LicenseConversionTask$SourceLicenseContext": "<p>Information about the license type this conversion task
converted from.</p>",
      "LicenseConversionTask$DestinationLicenseContext": "<p>Information about the license type this conversion
task converted to.</p>"
    }
  },
  "LicenseConversionTask": {
    "base": "<p>Information about a license type conversion task.</p>",
    "refs": {
      "LicenseConversionTasks$member": null
    }
  },
  "LicenseConversionTaskId": {
    "base": null,
    "refs": {
      "CreateLicenseConversionTaskForResourceResponse$LicenseConversionTaskId":
 "<p>The ID of the created license type conversion task.</p>",
      "GetLicenseConversionTaskRequest$LicenseConversionTaskId": "<p>ID of the license type conversion task to
retrieve information on.</p>",
      "GetLicenseConversionTaskResponse$LicenseConversionTaskId": "<p>ID of the license type conversion
task.</p>",
      "LicenseConversionTask$LicenseConversionTaskId": "<p>The ID of the license type conversion task.</p>"
    }
  },
  "LicenseConversionTaskStatus": {
    "base": null,
    "refs": {
      "GetLicenseConversionTaskResponse$Status": "<p>Status of the license type conversion task.</p>",
```

```
      "LicenseConversionTask$Status": "<p>The status of the conversion task.</p>"
    }
  },
  "LicenseConversionTasks": {
    "base": null,
    "refs": {
      "ListLicenseConversionTasksResponse$LicenseConversionTasks": "<p>Information about the license
configuration tasks for your account.</p>"
    }
  },
  "LicenseCountingType":
 {
    "base": null,
    "refs": {
      "CreateLicenseConfigurationRequest$LicenseCountingType": "<p>Dimension used to track the license
inventory.</p>",
      "GetLicenseConfigurationResponse$LicenseCountingType": "<p>Dimension for which the licenses are
counted.</p>",
      "LicenseConfiguration$LicenseCountingType": "<p>Dimension to use to track the license inventory.</p>"
    }
  },
  "LicenseDeletionStatus": {
    "base": null,
    "refs": {
      "DeleteLicenseResponse$Status": "<p>License status.</p>"
    }
  },
  "LicenseList": {
    "base": null,
    "refs": {
      "ListLicenseVersionsResponse$Licenses": "<p>License details.</p>",
      "ListLicensesResponse$Licenses": "<p>License details.</p>"
    }
  },
  "LicenseOperationFailure": {
    "base": "<p>Describes the failure of a license operation.</p>",
    "refs": {
      "LicenseOperationFailureList$member": null
    }
  },
  "LicenseOperationFailureList": {
    "base":
null,
    "refs": {
      "ListFailuresForLicenseConfigurationOperationsResponse$LicenseOperationFailureList": "<p>License
configuration operations that failed.</p>"
    }
  },
```

```
"LicenseSpecification": {
 "base": "<p>Details for associating a license configuration with a resource.</p>",
 "refs": {
   "LicenseSpecifications$member": null
 }
},
"LicenseSpecifications": {
 "base": null,
 "refs": {
   "ListLicenseSpecificationsForResourceResponse$LicenseSpecifications": "<p>License configurations
associated with a resource.</p>",
   "UpdateLicenseSpecificationsForResourceRequest$AddLicenseSpecifications": "<p>ARNs of the license
configurations to add.</p>",
   "UpdateLicenseSpecificationsForResourceRequest$RemoveLicenseSpecifications": "<p>ARNs of the license
configurations to remove.</p>"
 }
},
"LicenseStatus": {
 "base": null,
 "refs": {
   "CreateLicenseResponse$Status": "<p>License status.</p>",
   "CreateLicenseVersionRequest$Status": "<p>License status.</p>",
   "CreateLicenseVersionResponse$Status": "<p>License status.</p>",
   "GrantedLicense$Status": "<p>Granted license status.</p>",
   "License$Status": "<p>License status.</p>"
 }
},
"LicenseUsage": {
 "base": "<p>Describes the entitlement usage associated with a license.</p>",
 "refs": {
   "GetLicenseUsageResponse$LicenseUsage": "<p>License usage details.</p>"
 }
},
"LicenseUsageException": {
 "base": "<p>You do not have enough licenses available to support a new resource launch.</p>",
 "refs": {
 }
},
"ListAssociationsForLicenseConfigurationRequest": {
 "base": null,
 "refs": {
 }
},
"ListAssociationsForLicenseConfigurationResponse": {
 "base": null,
 "refs": {
 }
},
```

```
     "ListDistributedGrantsRequest": {
      "base": null,
      "refs": {
      }
     },
     "ListDistributedGrantsResponse":
{
      "base": null,
      "refs": {
      }
     },
     "ListFailuresForLicenseConfigurationOperationsRequest": {
      "base": null,
      "refs": {
      }
     },
     "ListFailuresForLicenseConfigurationOperationsResponse": {
      "base": null,
      "refs": {
      }
     },
     "ListLicenseConfigurationsRequest": {
      "base": null,
      "refs": {
      }
     },
     "ListLicenseConfigurationsResponse": {
      "base": null,
      "refs": {
      }
     },
     "ListLicenseConversionTasksRequest": {
      "base": null,
      "refs": {
      }
     },
     "ListLicenseConversionTasksResponse": {
      "base": null,
      "refs": {
      }
     },
     "ListLicenseManagerReportGeneratorsRequest": {
      "base": null,
      "refs": {
      }
     },
     "ListLicenseManagerReportGeneratorsResponse": {
      "base": null,
```

"refs": {
    }
  },
  "ListLicenseSpecificationsForResourceRequest": {
    "base": null,
    "refs": {
    }
  },
  "ListLicenseSpecificationsForResourceResponse":
{
    "base": null,
    "refs": {
    }
  },
  "ListLicenseVersionsRequest": {
    "base": null,
    "refs": {
    }
  },
  "ListLicenseVersionsResponse": {
    "base": null,
    "refs": {
    }
  },
  "ListLicensesRequest": {
    "base": null,
    "refs": {
    }
  },
  "ListLicensesResponse": {
    "base": null,
    "refs": {
    }
  },
  "ListReceivedGrantsForOrganizationRequest": {
    "base": null,
    "refs": {
    }
  },
  "ListReceivedGrantsForOrganizationResponse": {
    "base": null,
    "refs": {
    }
  },
  "ListReceivedGrantsRequest": {
    "base": null,
    "refs": {
    }

```
    },
    "ListReceivedGrantsResponse": {
     "base": null,
     "refs": {
     }
    },
    "ListReceivedLicensesForOrganizationRequest": {
     "base": null,
     "refs": {
     }
    },
    "ListReceivedLicensesForOrganizationResponse": {
     "base": null,
     "refs": {
     }
    },
    "ListReceivedLicensesRequest":
{
     "base": null,
     "refs": {
     }
    },
    "ListReceivedLicensesResponse": {
     "base": null,
     "refs": {
     }
    },
    "ListResourceInventoryRequest": {
     "base": null,
     "refs": {
     }
    },
    "ListResourceInventoryResponse": {
     "base": null,
     "refs": {
     }
    },
    "ListTagsForResourceRequest": {
     "base": null,
     "refs": {
     }
    },
    "ListTagsForResourceResponse": {
     "base": null,
     "refs": {
     }
    },
    "ListTokensRequest": {
```

```
   "base": null,
   "refs": {
   }
  },
  "ListTokensResponse": {
   "base": null,
   "refs": {
   }
  },
  "ListUsageForLicenseConfigurationRequest": {
   "base": null,
   "refs": {
   }
  },
  "ListUsageForLicenseConfigurationResponse": {
   "base": null,
   "refs": {
   }
  },
  "Location": {
   "base": null,
   "refs": {
    "RedirectException$Location": null
   }
  },
  "Long":
{
   "base": null,
   "refs": {
    "Entitlement$MaxCount": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
   }
  },
  "ManagedResourceSummary": {
   "base": "<p>Summary information about a managed resource.</p>",
   "refs": {
    "ManagedResourceSummaryList$member": null
   }
  },
  "ManagedResourceSummaryList": {
   "base": null,
   "refs": {
    "GetLicenseConfigurationResponse$ManagedResourceSummaryList": "<p>Summaries of the managed
resources.</p>",
    "LicenseConfiguration$ManagedResourceSummaryList": "<p>Summaries for managed resources.</p>"
   }
  },
  "MaxSize100": {
   "base": null,
```

```json
    "refs": {
      "ListDistributedGrantsRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListLicenseManagerReportGeneratorsRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListLicenseVersionsRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListLicensesRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListReceivedGrantsForOrganizationRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListReceivedGrantsRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListReceivedLicensesForOrganizationRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListReceivedLicensesRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>",
      "ListTokensRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>"
    }
  },
  "MaxSize3StringList": {
    "base": null,
    "refs": {
      "CreateTokenRequest$TokenProperties": "<p>Data specified by the caller to be included in the JWT token. The data is mapped to the amr claim of the JWT token.</p>",
      "GetAccessTokenRequest$TokenProperties": "<p>Token properties to validate against those present in the JWT token.</p>",
      "TokenData$TokenProperties": "<p>Data specified by the caller.</p>"
    }
  },
  "Message": {
    "base": null,
    "refs": {
      "AccessDeniedException$Message": null,
      "AuthorizationException$Message": null,
      "ConflictException$Message": null,
      "EntitlementNotAllowedException$Message": null,
      "FailedDependencyException$Message": null,
      "FilterLimitExceededException$Message": null,
      "InvalidParameterValueException$Message": null,
      "InvalidResourceStateException$Message": null,
      "LicenseUsageException$Message": null,
      "NoEntitlementsAllowedException$Message": null,
      "RateLimitExceededException$Message": null,
      "RedirectException$Message": null,
      "ResourceLimitExceededException$Message": null,
      "ResourceNotFoundException$Message": null,
      "ServerInternalException$Message": null,
      "UnsupportedDigitalSignatureMethodException$Message": null,
      "ValidationException$Message": null
```

```
      }
    },
    "Metadata": {
      "base": "<p>Describes key/value pairs.</p>",
      "refs": {
        "MetadataList$member": null
      }
    },
    "MetadataList": {
      "base": null,
      "refs": {
        "CheckoutBorrowLicenseRequest$CheckoutMetadata": "<p>Information about constraints.</p>",
        "CheckoutBorrowLicenseResponse$CheckoutMetadata": "<p>Information about constraints.</p>",
        "CreateLicenseRequest$LicenseMetadata": "<p>Information about the license.</p>",
        "CreateLicenseVersionRequest$LicenseMetadata": "<p>Information about the license.</p>",
        "GrantedLicense$LicenseMetadata": "<p>Granted license metadata.</p>",
        "License$LicenseMetadata": "<p>License metadata.</p>",
        "LicenseOperationFailure$MetadataList": "<p>Reserved.</p>"
      }
    },
    "NoEntitlementsAllowedException": {

      "base": "<p>There are no entitlements found for this license, or the entitlement maximum count is reached.</p>",
      "refs": {
      }
    },
    "Options": {
      "base": "<p>The options you can specify when you create a new version of a grant, such as activation override
behavior. For more information, see <a href=\"https://docs.aws.amazon.com/license-
manager/latest/userguide/granted-licenses.html\">Granted licenses in License Manager</a> in the <i>License
Manager User Guide</i>.</p>",
      "refs": {
        "CreateGrantVersionRequest$Options": "<p>The options specified for the grant.</p>",
        "Grant$Options": "<p>The options specified for the grant.</p>"
      }
    },
    "OrganizationConfiguration": {
      "base": "<p>Configuration information for Organizations.</p>",
      "refs": {
        "GetServiceSettingsResponse$OrganizationConfiguration": "<p>Indicates whether Organizations is integrated
with License Manager for cross-account discovery.</p>",
        "UpdateServiceSettingsRequest$OrganizationConfiguration":
"<p>Enables integration with Organizations for cross-account discovery.</p>"
      }
    },
    "PrincipalArnList": {
      "base": null,
      "refs": {
```

"CreateGrantRequest$Principals": "<p>The grant principals. You can specify one of the following as an Amazon Resource Name (ARN):</p>   <p>An Amazon Web Services account, which includes only the account specified.</p>   <p>An organizational unit (OU), which includes all accounts in the OU.</p>   <p>An organization, which will include all accounts across your organization.</p> "
    }
  },
  "ProductInformation": {
   "base": "<p>Describes product information for a license configuration.</p>",
   "refs": {
    "ProductInformationList$member": null
   }
  },
  "ProductInformationFilter": {
   "base": "<p>Describes product information filters.</p>",
   "refs": {
    "ProductInformationFilterList$member":
null
   }
  },
  "ProductInformationFilterList": {
   "base": null,
   "refs": {
    "ProductInformation$ProductInformationFilterList": "<p>A Product information filter consists of a <code>ProductInformationFilterComparator</code> which is a logical operator, a <code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a <code>ProductInformationFilterValue</code> that specifies the value to filter on. </p> <p>Accepted values for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for <code>ProductInformationFilterComparator</code>. </p> <p>The following filters and are supported when the resource type is <code>SSM_MANAGED</code>:</p>   <p> <code>Application Name</code> - The name of the application. Logical operator is <code>EQUALS</code>.</p>   <p> <code>Application Publisher</code> - The publisher of the application. Logical operator is <code>EQUALS</code>.</p>   <p> <code>Application Version</code> - The version of the application. Logical operator is <code>EQUALS</code>.</p>   <p> <code>Platform Name</code> - The name of the platform. Logical operator is <code>EQUALS</code>.</p>   <p> <code>Platform Type</code> - The platform type. Logical operator is <code>EQUALS</code>.</p>   <p> <code>Tag:key</code> - The key of a tag attached to an Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is <code>NOT_EQUALS</code>. The key for your tag must be appended to <code>Tag:</code> following the example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you are not using values for the key. </p>   <p> <code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from automated discovery. Logical operator is <code>NOT_EQUALS</code>.</p>   <p> <code>License Included</code> - The type of license included. Logical operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>. Possible values are: <code>sql-server-enterprise</code> | <code>sql-server-standard</code> | <code>sql-server-web</code> | <code>windows-server-datacenter</code>.</p>   <p>The following filters and logical operators are supported when the resource type is <code>RDS</code>:</p>   <p> <code>Engine Edition</code> - The edition of the database engine. Logical operator is <code>EQUALS</code>. Possible values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-se2</code>.</p>

  <p> <code>License Pack</code> - The license pack. Logical operator is <code>EQUALS</code>. Possible values are: <code>data guard</code> | <code>diagnostic pack sqlt</code> | <code>tuning pack sqlt</code> | <code>ols</code> | <code>olap</code>.</p> "
      }
    },
    "ProductInformationList": {
      "base": null,
      "refs": {
        "CreateLicenseConfigurationRequest$ProductInformationList":
"<p>Product information.</p>",
        "GetLicenseConfigurationResponse$ProductInformationList": "<p>Product information.</p>",
        "LicenseConfiguration$ProductInformationList": "<p>Product information.</p>",
        "UpdateLicenseConfigurationRequest$ProductInformationList": "<p>New product information.</p>"
      }
    },
    "ProvisionalConfiguration": {
      "base": "<p>Details about a provisional configuration.</p>",
      "refs": {
        "ConsumptionConfiguration$ProvisionalConfiguration": "<p>Details about a provisional configuration.</p>"
      }
    },
    "RateLimitExceededException": {
      "base": "<p>Too many requests have been submitted. Try again after a brief wait.</p>",
      "refs": {
      }
    },
    "ReceivedMetadata": {
      "base": "<p>Metadata associated with received licenses and grants.</p>",
      "refs": {
        "GrantedLicense$ReceivedMetadata": "<p>Granted license received metadata.</p>"
      }
    },
    "ReceivedStatus": {
      "base": null,
      "refs": {
        "ReceivedMetadata$ReceivedStatus": "<p>Received status.</p>"
      }
    },
    "RedirectException": {
      "base": "<p>This is not the correct Region for the resource. Try again.</p>",
      "refs": {
      }
    },
    "RejectGrantRequest": {
      "base": null,
      "refs": {
      }
    },

"RejectGrantResponse": {
  "base": null,
  "refs": {
  }
},
"RenewType": {
  "base": null,
  "refs": {
    "ConsumptionConfiguration$RenewType": "<p>Renewal frequency.</p>"
  }
},
"ReportContext": {
  "base": "<p>Details of the license configuration that this generator reports on.</p>",
  "refs": {
    "CreateLicenseManagerReportGeneratorRequest$ReportContext": "<p>Defines the type of license configuration the report generator tracks.</p>",
    "ReportGenerator$ReportContext": "<p>License configuration type for this generator.</p>",
    "UpdateLicenseManagerReportGeneratorRequest$ReportContext": "<p>The report context.</p>"
  }
},
"ReportFrequency": {
  "base": "<p>Details about how frequently reports are generated.</p>",
  "refs": {
    "CreateLicenseManagerReportGeneratorRequest$ReportFrequency": "<p>Frequency by which reports are generated. Reports can be generated daily, monthly, or weekly.</p>",
    "ReportGenerator$ReportFrequency": "<p>Details about how frequently reports are generated.</p>",
    "UpdateLicenseManagerReportGeneratorRequest$ReportFrequency": "<p>Frequency by which reports are generated.</p>"
  }
},
"ReportFrequencyType": {
  "base": null,
  "refs": {
    "ReportFrequency$period": "<p>Time period between each report. The period can be daily, weekly, or monthly.</p>"
  }
},
"ReportGenerator": {
  "base": "<p>Describe the details of a report generator.</p>",
  "refs": {
    "GetLicenseManagerReportGeneratorResponse$ReportGenerator": "<p>A report generator that creates periodic reports about your license configurations.</p>",
    "ReportGeneratorList$member": null
  }
},
"ReportGeneratorList": {
  "base": null,

    "refs": {

      "ListLicenseManagerReportGeneratorsResponse$ReportGenerators": "<p>A report generator that creates periodic reports about your license configurations.</p>"

    }

  },

  "ReportGeneratorName": {

    "base": null,

    "refs": {

      "CreateLicenseManagerReportGeneratorRequest$ReportGeneratorName": "<p>Name of the report generator.</p>",

      "UpdateLicenseManagerReportGeneratorRequest$ReportGeneratorName": "<p>Name of the report generator.</p>"

    }

  },

  "ReportType": {

    "base": null,

    "refs": {

      "ReportTypeList$member": null

    }

  },

  "ReportTypeList": {

    "base": null,

    "refs": {

      "CreateLicenseManagerReportGeneratorRequest$Type": "<p>Type of reports to generate. The following report types an

 be generated:</p>   <p>License configuration report - Reports the number and details of consumed licenses for a license configuration.</p>   <p>Resource report - Reports the tracked licenses and resource consumption for a license configuration.</p> ",

      "ReportGenerator$ReportType": "<p>Type of reports that are generated.</p>",

      "UpdateLicenseManagerReportGeneratorRequest$Type": "<p>Type of reports to generate. The following report types are supported:</p>   <p>License configuration report - Reports the number and details of consumed licenses for a license configuration.</p>   <p>Resource report - Reports the tracked licenses and resource consumption for a license configuration.</p> "

    }

  },

  "ResourceInventory": {

    "base": "<p>Details about a resource.</p>",

    "refs": {

      "ResourceInventoryList$member": null

    }

  },

  "ResourceInventoryList": {

    "base": null,

    "refs": {

      "ListResourceInventoryResponse$ResourceInventoryList": "<p>Information about the resources.</p>"

    }

  },

  "ResourceLimitExceededException": {

     "base": "<p>Your resource limits have been exceeded.</p>",
     "refs": {
     }
    },
    "ResourceNotFoundException": {
     "base": "<p>The resource cannot be found.</p>",
     "refs": {
     }
    },
    "ResourceType": {
     "base": null,
     "refs": {
      "ConsumedLicenseSummary$ResourceType": "<p>Resource type of the resource consuming a license.</p>",
      "LicenseConfigurationAssociation$ResourceType": "<p>Type of server resource.</p>",
      "LicenseConfigurationUsage$ResourceType": "<p>Type of resource.</p>",
      "LicenseOperationFailure$ResourceType": "<p>Resource type.</p>",
      "ManagedResourceSummary$ResourceType": "<p>Type of resource associated with a license.</p>",
      "ResourceInventory$ResourceType": "<p>Type of resource.</p>"
     }
    },
    "S3Location": {
     "base":
   "<p>Details of the S3 bucket that report generator reports are published to.</p>",
     "refs": {
      "ReportGenerator$S3Location": "<p>Details of the S3 bucket that report generator reports are published
   to.</p>"
     }
    },
    "ServerInternalException": {
     "base": "<p>The server experienced an internal error. Try again.</p>",
     "refs": {
     }
    },
    "SignedToken": {
     "base": null,
     "refs": {
      "CheckoutBorrowLicenseResponse$SignedToken": "<p>Signed token.</p>",
      "CheckoutLicenseResponse$SignedToken": "<p>Signed token.</p>"
     }
    },
    "StatusReasonMessage": {
     "base": null,
     "refs": {
      "CreateGrantVersionRequest$StatusReason": "<p>Grant status reason.</p>",
      "DeleteGrantRequest$StatusReason": "<p>The Status reason for the delete request.</p>",
      "Grant$StatusReason": "<p>Grant status reason.</p>",
      "ReceivedMetadata$ReceivedStatusReason": "<p>Received status reason.</p>"
     }

```
        },

      "String": {
        "base": null,
        "refs": {
          "AcceptGrantResponse$Version": "<p>Grant version.</p>",
          "CheckInLicenseRequest$LicenseConsumptionToken": "<p>License consumption token.</p>",
          "CheckInLicenseRequest$Beneficiary": "<p>License beneficiary.</p>",
          "CheckoutBorrowLicenseRequest$NodeId": "<p>Node ID.</p>",
          "CheckoutBorrowLicenseResponse$LicenseConsumptionToken": "<p>License consumption token.</p>",
          "CheckoutBorrowLicenseResponse$NodeId": "<p>Node ID.</p>",
          "CheckoutLicenseRequest$ProductSKU": "<p>Product SKU.</p>",
          "CheckoutLicenseRequest$KeyFingerprint": "<p>Key fingerprint identifying the license.</p>",
          "CheckoutLicenseRequest$Beneficiary": "<p>License beneficiary.</p>",
          "CheckoutLicenseRequest$NodeId": "<p>Node ID.</p>",
          "CheckoutLicenseResponse$LicenseConsumptionToken": "<p>License consumption token.</p>",
          "CheckoutLicenseResponse$NodeId": "<p>Node ID.</p>",
          "CheckoutLicenseResponse$LicenseArn":
 "<p>Amazon Resource Name (ARN) of the checkout license.</p>",
          "CreateGrantRequest$GrantName": "<p>Grant name.</p>",
          "CreateGrantRequest$HomeRegion": "<p>Home Region of the grant.</p>",
          "CreateGrantResponse$Version": "<p>Grant version.</p>",
          "CreateGrantVersionRequest$GrantName": "<p>Grant name.</p>",
          "CreateGrantVersionRequest$SourceVersion": "<p>Current version of the grant.</p>",
          "CreateGrantVersionResponse$Version": "<p>New version of the grant.</p>",
          "CreateLicenseConfigurationRequest$Name": "<p>Name of the license configuration.</p>",
          "CreateLicenseConfigurationRequest$Description": "<p>Description of the license configuration.</p>",
          "CreateLicenseConfigurationResponse$LicenseConfigurationArn": "<p>Amazon Resource Name (ARN) of the
license configuration.</p>",
          "CreateLicenseManagerReportGeneratorRequest$Description": "<p>Description of the report generator.</p>",
          "CreateLicenseManagerReportGeneratorResponse$LicenseManagerReportGeneratorArn":
 "<p>The Amazon Resource Name (ARN) of the new report generator.</p>",
          "CreateLicenseRequest$LicenseName": "<p>License name.</p>",
          "CreateLicenseRequest$ProductName": "<p>Product name.</p>",
          "CreateLicenseRequest$ProductSKU": "<p>Product SKU.</p>",
          "CreateLicenseRequest$HomeRegion": "<p>Home Region for the license.</p>",
          "CreateLicenseRequest$Beneficiary": "<p>License beneficiary.</p>",
          "CreateLicenseResponse$Version": "<p>License version.</p>",
          "CreateLicenseVersionRequest$LicenseName": "<p>License name.</p>",
          "CreateLicenseVersionRequest$ProductName": "<p>Product name.</p>",
          "CreateLicenseVersionRequest$HomeRegion": "<p>Home Region of the license.</p>",
          "CreateLicenseVersionRequest$SourceVersion": "<p>Current version of the license.</p>",
          "CreateLicenseVersionResponse$Version": "<p>New version of the license.</p>",
          "CreateTokenResponse$TokenId":
 "<p>Token ID.</p>",
          "DeleteGrantRequest$Version": "<p>Current version of the grant.</p>",
          "DeleteGrantResponse$Version": "<p>Grant version.</p>",
          "DeleteLicenseConfigurationRequest$LicenseConfigurationArn": "<p>ID of the license configuration.</p>",
```

"DeleteLicenseManagerReportGeneratorRequest$LicenseManagerReportGeneratorArn": "<p>Amazon Resource Name (ARN) of the report generator to be deleted.</p>",
    "DeleteLicenseRequest$SourceVersion": "<p>Current version of the license.</p>",
    "DeleteTokenRequest$TokenId": "<p>Token ID.</p>",
    "Entitlement$Name": "<p>Entitlement name.</p>",
    "Entitlement$Value": "<p>Entitlement resource. Use only if the unit is None.</p>",
    "EntitlementData$Name": "<p>Entitlement data name.</p>",
    "EntitlementData$Value": "<p>Entitlement data value.</p>",
    "EntitlementUsage$Name": "<p>Entitlement usage name.</p>",
    "EntitlementUsage$ConsumedValue":
 "<p>Resource usage consumed.</p>",
    "EntitlementUsage$MaxCount": "<p>Maximum entitlement usage count.</p>",
    "ExtendLicenseConsumptionRequest$LicenseConsumptionToken": "<p>License consumption token.</p>",
    "ExtendLicenseConsumptionResponse$LicenseConsumptionToken": "<p>License consumption token.</p>",
    "FailedDependencyException$ErrorCode": null,
    "GetGrantRequest$Version": "<p>Grant version.</p>",
    "GetLicenseConfigurationRequest$LicenseConfigurationArn": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
    "GetLicenseConfigurationResponse$LicenseConfigurationId": "<p>Unique ID for the license configuration.</p>",
    "GetLicenseConfigurationResponse$LicenseConfigurationArn": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
    "GetLicenseConfigurationResponse$Name": "<p>Name of the license configuration.</p>",
    "GetLicenseConfigurationResponse$Description": "<p>Description of the license configuration.</p>",
    "GetLicenseConfigurationResponse$Status": "<p>License configuration status.</p>",
    "GetLicenseConfigurationResponse$OwnerAccountId": "<p>Account ID of the owner of the license configuration.</p>",
    "GetLicenseConversionTaskResponse$ResourceArn": "<p>Amazon Resource Names (ARN) of the resources the license conversion task is associated with.</p>",
    "GetLicenseConversionTaskResponse$StatusMessage": "<p>The status message for the conversion task.</p>",
    "GetLicenseManagerReportGeneratorRequest$LicenseManagerReportGeneratorArn": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
    "GetLicenseRequest$Version": "<p>License version.</p>",
    "GetServiceSettingsResponse$S3BucketArn": "<p>Regional S3 bucket path for storing reports, license trail event data, discovery data, and so on.</p>",
    "GetServiceSettingsResponse$SnsTopicArn": "<p>SNS topic configured to receive notifications from License Manager.</p>",
    "GetServiceSettingsResponse$LicenseManagerResourceShareArn": "<p>Amazon Resource Name (ARN) of the resource share. The License Manager management account provides member accounts with access to this share.</p>",
    "Grant$GrantName": "<p>Grant name.</p>",
    "Grant$HomeRegion": "<p>Home Region of the grant.</p>",
    "Grant$Version": "<p>Grant version.</p>",
    "GrantedLicense$LicenseName": "<p>License name.</p>",
    "GrantedLicense$ProductName": "<p>Product name.</p>",
    "GrantedLicense$ProductSKU": "<p>Product SKU.</p>",
    "GrantedLicense$HomeRegion": "<p>Home Region of the granted license.</p>",

    "GrantedLicense$Beneficiary": "<p>Granted license beneficiary.</p>",
    "GrantedLicense$Version": "<p>Version of the granted license.</p>",
    "InventoryFilter$Name": "<p>Name of the filter.</p>",
    "InventoryFilter$Value": "<p>Value of the filter.</p>",
    "Issuer$Name": "<p>Issuer name.</p>",
    "Issuer$SignKey":
 "<p>Asymmetric KMS key from Key Management Service. The KMS key must have a key usage of sign and verify, and support the RSASSA-PSS SHA-256 signing algorithm.</p>",
    "IssuerDetails$Name": "<p>Issuer name.</p>",
    "IssuerDetails$SignKey": "<p>Asymmetric KMS key from Key Management Service. The KMS key must have a key usage of sign and verify, and support the RSASSA-PSS SHA-256 signing algorithm.</p>",
    "IssuerDetails$KeyFingerprint": "<p>Issuer key fingerprint.</p>",
    "License$LicenseName": "<p>License name.</p>",
    "License$ProductName": "<p>Product name.</p>",
    "License$ProductSKU": "<p>Product SKU.</p>",
    "License$HomeRegion": "<p>Home Region of the license.</p>",
    "License$Beneficiary": "<p>License beneficiary.</p>",
    "License$Version": "<p>License version.</p>",
    "LicenseConfiguration$LicenseConfigurationId": "<p>Unique ID of the license configuration.</p>",
    "LicenseConfiguration$LicenseConfigurationArn":
 "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
    "LicenseConfiguration$Name": "<p>Name of the license configuration.</p>",
    "LicenseConfiguration$Description": "<p>Description of the license configuration.</p>",
    "LicenseConfiguration$Status": "<p>Status of the license configuration.</p>",
    "LicenseConfiguration$OwnerAccountId": "<p>Account ID of the license configuration's owner.</p>",
    "LicenseConfigurationAssociation$ResourceArn": "<p>Amazon Resource Name (ARN) of the resource.</p>",
    "LicenseConfigurationAssociation$ResourceOwnerId": "<p>ID of the Amazon Web Services account that owns the resource consuming licenses.</p>",
    "LicenseConfigurationAssociation$AmiAssociationScope": "<p>Scope of AMI associations. The possible value is <code>cross-account</code>.</p>",
    "LicenseConfigurationUsage$ResourceArn": "<p>Amazon Resource Name (ARN) of the resource.</p>",
    "LicenseConfigurationUsage$ResourceStatus":
 "<p>Status of the resource.</p>",
    "LicenseConfigurationUsage$ResourceOwnerId": "<p>ID of the account that owns the resource.</p>",
    "LicenseConversionTask$ResourceArn": "<p>The Amazon Resource Name (ARN) of the resource associated with the license type conversion task.</p>",
    "LicenseConversionTask$StatusMessage": "<p>The status message for the conversion task.</p>",
    "LicenseOperationFailure$ResourceArn": "<p>Amazon Resource Name (ARN) of the resource.</p>",
    "LicenseOperationFailure$ErrorMessage": "<p>Error message.</p>",
    "LicenseOperationFailure$OperationName": "<p>Name of the operation.</p>",
    "LicenseOperationFailure$ResourceOwnerId": "<p>ID of the Amazon Web Services account that owns the resource.</p>",
    "LicenseOperationFailure$OperationRequestedBy": "<p>The requester is \"License Manager Automated Discovery\".</p>",
    "LicenseSpecification$LicenseConfigurationArn": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
    "LicenseSpecification$AmiAssociationScope": "<p>Scope of AMI associations. The possible value is <code>cross-account</code>.</p>",

"ListAssociationsForLicenseConfigurationRequest$LicenseConfigurationArn": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
    "ListAssociationsForLicenseConfigurationRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListAssociationsForLicenseConfigurationResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListDistributedGrantsRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListDistributedGrantsResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListFailuresForLicenseConfigurationOperationsRequest$LicenseConfigurationArn": "<p>Amazon Resource Name of the license configuration.</p>",
    "ListFailuresForLicenseConfigurationOperationsRequest$NextToken": "<p>Token for the next set of results.</p>",

 "ListFailuresForLicenseConfigurationOperationsResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicenseConfigurationsRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicenseConfigurationsResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicenseConversionTasksRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicenseConversionTasksResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicenseManagerReportGeneratorsRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicenseManagerReportGeneratorsResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicenseSpecificationsForResourceRequest$ResourceArn": "<p>Amazon Resource Name (ARN) of a resource that has an associated license configuration.</p>",
    "ListLicenseSpecificationsForResourceRequest$NextToken": "<p>Token for the next set of results.</p>",
     "ListLicenseSpecificationsForResourceResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicenseVersionsRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicenseVersionsResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicensesRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListLicensesResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListReceivedGrantsForOrganizationRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListReceivedGrantsForOrganizationResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListReceivedGrantsRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListReceivedGrantsResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListReceivedLicensesForOrganizationRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListReceivedLicensesForOrganizationResponse$NextToken":
 "<p>Token for the next set of results.</p>",
    "ListReceivedLicensesRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListReceivedLicensesResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListResourceInventoryRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListResourceInventoryResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListTagsForResourceRequest$ResourceArn": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
    "ListTokensRequest$NextToken": "<p>Token for the next set of results.</p>",
    "ListTokensResponse$NextToken": "<p>Token for the next set of results.</p>",
    "ListUsageForLicenseConfigurationRequest$LicenseConfigurationArn": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
    "ListUsageForLicenseConfigurationRequest$NextToken": "<p>Token for the next set of results.</p>",
     "ListUsageForLicenseConfigurationResponse$NextToken": "<p>Token for the next set of results.</p>",
    "MaxSize3StringList$member": null,
    "Metadata$Name": "<p>The key name.</p>",

"Metadata$Value": "<p>The value.</p>",
    "ProductInformation$ResourceType": "<p>Resource type. The possible values are
<code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
    "ProductInformationFilter$ProductInformationFilterName": "<p>Filter name.</p>",
    "ProductInformationFilter$ProductInformationFilterComparator": "<p>Logical operator.</p>",
    "RejectGrantResponse$Version": "<p>Grant version.</p>",
    "ReportGenerator$ReportGeneratorName": "<p>Name of the report generator.</p>",
    "ReportGenerator$LicenseManagerReportGeneratorArn": "<p>Amazon Resource Name (ARN) of the report
generator.</p>",
    "ReportGenerator$LastRunStatus": "<p>Status of the last report generation attempt.</p>",
    "ReportGenerator$LastRunFailureReason": "<p>Failure
 message for the last report generation attempt.</p>",
    "ReportGenerator$LastReportGenerationTime": "<p>Time the last report was generated at.</p>",
    "ReportGenerator$ReportCreatorAccount": "<p>The Amazon Web Services account ID used to create the
report generator.</p>",
    "ReportGenerator$Description": "<p>Description of the report generator.</p>",
    "ReportGenerator$CreateTime": "<p>Time the report was created.</p>",
    "ResourceInventory$ResourceId": "<p>ID of the resource.</p>",
    "ResourceInventory$ResourceArn": "<p>Amazon Resource Name (ARN) of the resource.</p>",
    "ResourceInventory$Platform": "<p>Platform of the resource.</p>",
    "ResourceInventory$PlatformVersion": "<p>Platform version of the resource in the inventory.</p>",
    "ResourceInventory$ResourceOwningAccountId": "<p>ID of the account that owns the resource.</p>",
    "S3Location$bucket": "<p>Name of the S3 bucket reports are published to.</p>",

 "S3Location$keyPrefix": "<p>Prefix of the S3 bucket reports are published to.</p>",
    "StringList$member": null,
    "Tag$Key": "<p>Tag key.</p>",
    "Tag$Value": "<p>Tag value.</p>",
    "TagKeyList$member": null,
    "TagResourceRequest$ResourceArn": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
    "TokenData$TokenId": "<p>Token ID.</p>",
    "TokenData$TokenType": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>",
    "TokenData$LicenseArn": "<p>Amazon Resource Name (ARN) of the license.</p>",
    "TokenData$Status": "<p>Token status. The possible values are <code>AVAILABLE</code> and
<code>DELETED</code>.</p>",
    "UntagResourceRequest$ResourceArn": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
    "UpdateLicenseConfigurationRequest$LicenseConfigurationArn": "<p>Amazon Resource Name (ARN) of the
license configuration.</p>",
    "UpdateLicenseConfigurationRequest$Name":
 "<p>New name of the license configuration.</p>",
    "UpdateLicenseConfigurationRequest$Description": "<p>New description of the license configuration.</p>",
    "UpdateLicenseManagerReportGeneratorRequest$LicenseManagerReportGeneratorArn": "<p>Amazon
Resource Name (ARN) of the report generator to update.</p>",
    "UpdateLicenseManagerReportGeneratorRequest$Description": "<p>Description of the report generator.</p>",
    "UpdateLicenseSpecificationsForResourceRequest$ResourceArn": "<p>Amazon Resource Name (ARN) of the
Amazon Web Services resource.</p>",

```
    "UpdateServiceSettingsRequest$S3BucketArn": "<p>Amazon Resource Name (ARN) of the Amazon S3
bucket where the License Manager information is stored.</p>",
    "UpdateServiceSettingsRequest$SnsTopicArn": "<p>Amazon Resource Name (ARN) of the Amazon SNS
topic used for License Manager alerts.</p>"
    }
  },
  "StringList": {
    "base": null,
    "refs": {
      "CreateLicenseConfigurationRequest$LicenseRules":
 "<p>License rules. The syntax is #name=value (for example, #allowedTenancy=EC2-DedicatedHost). The
available rules vary by dimension, as follows.</p> <ul> <li> <p> <code>Cores</code> dimension:
<code>allowedTenancy</code> | <code>licenseAffinityToHost</code> | <code>maximumCores</code> |
<code>minimumCores</code> </p> </li> <li> <p> <code>Instances</code> dimension:
<code>allowedTenancy</code> | <code>maximumCores</code> | <code>minimumCores</code> |
<code>maximumSockets</code> | <code>minimumSockets</code> | <code>maximumVcpus</code> |
<code>minimumVcpus</code> </p> </li> <li> <p> <code>Sockets</code> dimension:
<code>allowedTenancy</code> | <code>licenseAffinityToHost</code> | <code>maximumSockets</code> |
<code>minimumSockets</code> </p> </li> <li> <p> <code>vCPUs</code> dimension:
<code>allowedTenancy</code> | <code>honorVcpuOptimization</code> | <code>maximumVcpus</code> |
<code>minimumVcpus</code> </p> </li> </ul> <p>The unit for <code>licenseAffinityToHost</code>
 is days and the range is 1 to 180. The possible values for <code>allowedTenancy</code> are <code>EC2-
Default</code>, <code>EC2-DedicatedHost</code>, and <code>EC2-DedicatedInstance</code>. The possible
values for <code>honorVcpuOptimization</code> are <code>True</code> and <code>False</code>.</p>",
    "GetLicenseConfigurationResponse$LicenseRules": "<p>License rules.</p>",
    "LicenseConfiguration$LicenseRules": "<p>License rules.</p>",
    "ListLicenseConfigurationsRequest$LicenseConfigurationArns": "<p>Amazon Resource Names (ARN) of the
license configurations.</p>",
    "ListTokensRequest$TokenIds": "<p>Token IDs.</p>",
    "ProductInformationFilter$ProductInformationFilterValue": "<p>Filter value.</p>",
    "UpdateLicenseConfigurationRequest$LicenseRules": "<p>New license rule. The only rule that you can add
after you create a license configuration is licenseAffinityToHost.</p>"
    }
  },
  "Tag": {
    "base": "<p>Details about a tag
 for a license configuration.</p>",
    "refs": {
      "TagList$member": null
    }
  },
  "TagKeyList": {
    "base": null,
    "refs": {
      "UntagResourceRequest$TagKeys": "<p>Keys identifying the tags to remove.</p>"
    }
  },
  "TagList": {
```

```
    "base": null,
    "refs": {
      "CreateLicenseConfigurationRequest$Tags": "<p>Tags to add to the license configuration.</p>",
      "CreateLicenseManagerReportGeneratorRequest$Tags": "<p>Tags to add to the report generator.</p>",
      "GetLicenseConfigurationResponse$Tags": "<p>Tags for the license configuration.</p>",
      "ListTagsForResourceResponse$Tags": "<p>Information about the tags.</p>",
      "ReportGenerator$Tags": "<p>Tags associated with the report generator.</p>",
      "TagResourceRequest$Tags": "<p>One or more tags.</p>"
    }
  },
  "TagResourceRequest": {
    "base": null,
    "refs": {
    }
  },
  "TagResourceResponse": {
    "base": null,
    "refs": {

    }
  },
  "TokenData": {
    "base": "<p>Describes a token.</p>",
    "refs": {
      "TokenList$member": null
    }
  },
  "TokenList": {
    "base": null,
    "refs": {
      "ListTokensResponse$Tokens": "<p>Received token details.</p>"
    }
  },
  "TokenString": {
    "base": null,
    "refs": {
      "CreateTokenResponse$Token": "<p>Refresh token, encoded as a JWT token.</p>",
      "GetAccessTokenRequest$Token": "<p>Refresh token, encoded as a JWT token.</p>",
      "GetAccessTokenResponse$AccessToken": "<p>Temporary access token.</p>"
    }
  },
  "TokenType": {
    "base": null,
    "refs": {
      "CreateTokenResponse$TokenType": "<p>Token type.</p>"
    }
  },
  "UnsupportedDigitalSignatureMethodException": {
```

```json
      "base": "<p>The digital signature method is unsupported. Try your request again.</p>",
      "refs": {
      }
    },
    "UntagResourceRequest": {
      "base": null,
      "refs": {
      }
    },
    "UntagResourceResponse":
{
      "base": null,
      "refs": {
      }
    },
    "UpdateLicenseConfigurationRequest": {
      "base": null,
      "refs": {
      }
    },
    "UpdateLicenseConfigurationResponse": {
      "base": null,
      "refs": {
      }
    },
    "UpdateLicenseManagerReportGeneratorRequest": {
      "base": null,
      "refs": {
      }
    },
    "UpdateLicenseManagerReportGeneratorResponse": {
      "base": null,
      "refs": {
      }
    },
    "UpdateLicenseSpecificationsForResourceRequest": {
      "base": null,
      "refs": {
      }
    },
    "UpdateLicenseSpecificationsForResourceResponse": {
      "base": null,
      "refs": {
      }
    },
    "UpdateServiceSettingsRequest": {
      "base": null,
      "refs": {
```

```
      }
    },
    "UpdateServiceSettingsResponse": {
     "base": null,
     "refs": {
     }
    },
    "UsageOperation": {
     "base": null,
     "refs": {
       "LicenseConversionContext$UsageOperation": "<p>The Usage operation value
 that corresponds to the license type you are converting your resource from. For more information about which
platforms correspond to which usage operation values see <a
href=\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-fields.html#billing-info\">Sample
data: usage operation by platform </a> </p>"
     }
    },
    "ValidationException": {
     "base": "<p>The provided input is not valid. Try your request again.</p>",
     "refs": {
     }
    }
   }
}
{
 "version": "2.0",
 "service": "<p>With License Manager, you can discover and track your commercial Linux subscriptions on running
Amazon EC2 instances.</p>",
 "operations": {
   "GetServiceSettings": "<p>Lists the Linux subscriptions service settings.</p>",
   "ListLinuxSubscriptionInstances": "<p>Lists the running Amazon EC2 instances that were discovered with
commercial Linux subscriptions.</p>",
   "ListLinuxSubscriptions": "<p>Lists the Linux subscriptions that have been discovered. If you have linked your
organization, the returned results will include data aggregated across your accounts in Organizations.</p>",
   "UpdateServiceSettings": "<p>Updates the service settings for Linux subscriptions.</p>"
 },
 "shapes": {
   "Boolean": {
     "base": null,
     "refs": {
       "UpdateServiceSettingsRequest$AllowUpdate": "<p>Describes if updates are allowed to the service settings for
Linux subscriptions. If you allow updates, you can aggregate Linux subscription data in
 more than one home Region.</p>"
     }
   },
   "BoxInteger": {
     "base": null,
     "refs": {
```

          "ListLinuxSubscriptionInstancesRequest$MaxResults": "<p>Maximum number of results to return in a single
call.</p>",
          "ListLinuxSubscriptionsRequest$MaxResults": "<p>Maximum number of results to return in a single call.</p>"
        }
      },
      "BoxLong": {
        "base": null,
        "refs": {
          "Subscription$InstanceCount": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "Filter": {
        "base": "<p>A filter object that is used to return more specific results from a describe operation. Filters can be
used to match a set of resources by specific criteria.</p>",
        "refs": {
          "FilterList$member": null
        }
      },
      "FilterList": {
        "base": null,
        "refs": {
          "ListLinuxSubscriptionInstancesRequest$Filters": "<p>An array of structures that you can use to filter the
results to those that match one
 or more sets of key-value pairs that you specify. For example, you can filter by the name of <code>AmiID</code>
with an optional operator to see subscriptions that match, partially match, or don't match a certain Amazon Machine
Image (AMI) ID.</p> <p>The valid names for this filter are:</p>   <p> <code>AmiID</code> </p>
 <p> <code>InstanceID</code> </p>   <p> <code>AccountID</code> </p>   <p>
<code>Status</code> </p>   <p> <code>Region</code> </p>   <p>
<code>UsageOperation</code> </p>   <p> <code>ProductCode</code> </p>   <p>
<code>InstanceType</code> </p>   <p>The valid Operators for this filter are:</p>   <p>
<code>contains</code> </p>   <p> <code>equals</code> </p>   <p> <code>Notequal</code>
</p> ",
          "ListLinuxSubscriptionsRequest$Filters": "<p>An array of structures that you can use to filter the results to
those that match one or more sets of key-value
 pairs that you specify. For example, you can filter by the name of <code>Subscription</code> with an optional
operator to see subscriptions that match, partially match, or don't match a certain subscription's name.</p> <p>The
valid names for this filter are:</p>   <p> <code>Subscription</code> </p>   <p>The valid
Operators for this filter are:</p>   <p> <code>contains</code> </p>   <p> <code>equals</code>
</p>   <p> <code>Notequal</code> </p> "
        }
      },
      "GetServiceSettingsRequest": {
        "base": null,
        "refs": {
        }
      },
      "GetServiceSettingsResponse": {
        "base": null,

```
      "refs": {
      }
    },
    "Instance": {
      "base": "<p>Details discovered information about a running instance using Linux subscriptions.</p>",
      "refs": {
        "InstanceList$member": null
      }
    },
    "InstanceList": {
      "base": null,
      "refs": {
        "ListLinuxSubscriptionInstancesResponse$Instances":
"<p>An array that contains instance objects.</p>"
      }
    },
    "InternalServerException": {
      "base": "<p>An exception occurred with the service.</p>",
      "refs": {
      }
    },
    "LinuxSubscriptionsDiscovery": {
      "base": null,
      "refs": {
        "GetServiceSettingsResponse$LinuxSubscriptionsDiscovery": "<p>Lists if discovery has been enabled for
Linux subscriptions.</p>",
        "UpdateServiceSettingsRequest$LinuxSubscriptionsDiscovery": "<p>Describes if the discovery of Linux
subscriptions is enabled.</p>",
        "UpdateServiceSettingsResponse$LinuxSubscriptionsDiscovery": "<p>Lists if discovery has been enabled for
Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "base": "<p>Lists the settings defined for discovering Linux subscriptions.</p>",
      "refs": {
        "GetServiceSettingsResponse$LinuxSubscriptionsDiscoverySettings": "<p>Lists the settings defined for Linux
subscriptions discovery. The settings
 include if Organizations integration has been enabled, and which Regions data will be aggregated from.</p>",
        "UpdateServiceSettingsRequest$LinuxSubscriptionsDiscoverySettings": "<p>The settings defined for Linux
subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data
will be aggregated from.</p>",
        "UpdateServiceSettingsResponse$LinuxSubscriptionsDiscoverySettings": "<p>The settings defined for Linux
subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data
will be aggregated from.</p>"
      }
    },
    "ListLinuxSubscriptionInstancesRequest": {
      "base": "<p>NextToken length limit is half of ddb accepted limit. Increase this limit if parameters in request
```

increases.</p>",
    "refs": {
    }
  },
  "ListLinuxSubscriptionInstancesRequestNextTokenString": {
    "base": null,
    "refs": {
      "ListLinuxSubscriptionInstancesRequest$NextToken":
 "<p>Token for the next set of results.</p>"
    }
  },
  "ListLinuxSubscriptionInstancesResponse": {
    "base": null,
    "refs": {
    }
  },
  "ListLinuxSubscriptionsRequest": {
    "base": "<p>NextToken length limit is half of ddb accepted limit. Increase this limit if parameters in request
increases.</p>",
    "refs": {
    }
  },
  "ListLinuxSubscriptionsRequestNextTokenString": {
    "base": null,
    "refs": {
      "ListLinuxSubscriptionsRequest$NextToken": "<p>Token for the next set of results.</p>"
    }
  },
  "ListLinuxSubscriptionsResponse": {
    "base": null,
    "refs": {
    }
  },
  "Operator": {
    "base": null,
    "refs": {
      "Filter$Operator": "<p>An operator for filtering results.</p>"
    }
  },
  "OrganizationIntegration": {
    "base": null,
    "refs": {
      "LinuxSubscriptionsDiscoverySettings$OrganizationIntegration": "<p>Details if you have enabled resource
discovery across
 your accounts in Organizations.</p>"
    }
  },
  "ProductCodeList": {

      "base": null,
      "refs": {
        "Instance$ProductCode": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage operation values in the <i>License Manager User Guide</i> .</p>"
      }
    },
    "Status": {
      "base": null,
      "refs": {
        "GetServiceSettingsResponse$Status": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>",
        "UpdateServiceSettingsResponse$Status": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "String": {
      "base": null,
      "refs": {
        "Filter$Name": "<p>The type of name to filter by.</p>",
        "Instance$AccountID": "<p>The account ID which owns the instance.</p>",
        "Instance$AmiId": "<p>The AMI ID used to launch the instance.</p>",
        "Instance$InstanceID": "<p>The instance ID of the resource.</p>",
        "Instance$InstanceType": "<p>The instance type of the resource.</p>",
        "Instance$LastUpdatedTime": "<p>The time in which the last discovery updated the instance details.</p>",
        "Instance$Region": "<p>The Region the instance is running in.</p>",
        "Instance$Status": "<p>The status of the instance.</p>",
        "Instance$SubscriptionName": "<p>The name of the subscription being used by the instance.</p>",
        "Instance$UsageOperation": "<p>The usage operation of the instance. For more information, see For more
information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-
usage-operation.html\">Usage operation values in the <i>License Manager User Guide</i>.</p>",
        "InternalServerException$message": null,
        "ListLinuxSubscriptionInstancesResponse$NextToken": "<p>Token for the next set of results.</p>",
        "ListLinuxSubscriptionsResponse$NextToken":
 "<p>Token for the next set of results.</p>",
        "ProductCodeList$member": null,
        "StringMap$key": null,
        "StringMap$value": null,
        "Subscription$Name": "<p>The name of the subscription.</p>",
        "Subscription$Type": "<p>The type of subscription. The type can be subscription-included with Amazon EC2,
Bring Your Own Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain
subscriptions may use licensing from the Amazon Web Services Marketplace as well as OS licensing from Amazon
EC2 or BYOS.</p>",
        "ThrottlingException$message": null,
        "ValidationException$message": null
      }
    },
    "StringList": {

```
    "base": null,
    "refs": {
      "Filter$Values": "<p>One or more values for the name to filter by.</p>",
      "GetServiceSettingsResponse$HomeRegions": "<p>The Region in which License Manager displays the
aggregated data for Linux subscriptions.</p>",
      "LinuxSubscriptionsDiscoverySettings$SourceRegions":
 "<p>The Regions in which to discover data for Linux subscriptions.</p>",
      "UpdateServiceSettingsResponse$HomeRegions": "<p>The Region in which License Manager displays the
aggregated data for Linux subscriptions.</p>"
    }
  },
  "StringListMemberString": {
    "base": null,
    "refs": {
      "StringList$member": null
    }
  },
  "StringMap": {
    "base": null,
    "refs": {
      "GetServiceSettingsResponse$StatusMessage": "<p>A message which details the Linux subscriptions service
settings current status.</p>",
      "UpdateServiceSettingsResponse$StatusMessage": "<p>A message which details the Linux subscriptions
service settings current status.</p>"
    }
  },
  "Subscription": {
    "base": "<p>An object which details a discovered Linux subscription.</p>",
    "refs": {
      "SubscriptionList$member": null
    }
  },
  "SubscriptionList": {
    "base": null,
    "refs": {

  "ListLinuxSubscriptionsResponse$Subscriptions": "<p>An array that contains subscription objects.</p>"
    }
  },
  "ThrottlingException": {
    "base": "<p>The request was denied due to request throttling.</p>",
    "refs": {
    }
  },
  "UpdateServiceSettingsRequest": {
    "base": null,
    "refs": {
    }
```

```
        },
      "UpdateServiceSettingsResponse": {
        "base": null,
        "refs": {
        }
      },
      "ValidationException": {
        "base": "<p>The provided input is not valid. Try your request again.</p>",
        "refs": {
        }
      }
    }
  }
}
{
  "testCases": [
    {
      "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.af-south-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "af-south-1"
      }
    },
    {
      "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-east-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "ap-east-1"
      }
    },
    {
      "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",

      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-1.amazonaws.com"
        }
```

```
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "ap-northeast-1"
      }
    },
    {
      "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-2.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "ap-northeast-2"
      }
    },
    {
      "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-northeast-3.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "ap-northeast-3"
      }
    },
    {
      "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-south-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "ap-south-1"
      }
    },
    {
      "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
```

```json
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-southeast-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack":
false,
        "Region": "ap-southeast-1"
      }
    },
    {
      "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-southeast-2.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "ap-southeast-2"
      }
    },
    {
      "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ap-southeast-3.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "ap-southeast-3"
      }
    },
    {
      "documentation":
"For region ca-central-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.ca-central-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
```

```
                    "UseDualStack": false,
                    "Region": "ca-central-1"
                }
            },
            {
                "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.eu-central-1.amazonaws.com"
                    }
                },
                "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region": "eu-central-1"
                }
            },
            {
                "documentation": "For region eu-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint":
{

                        "url": "https://license-manager.eu-north-1.amazonaws.com"
                    }
                },
                "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region": "eu-north-1"
                }
            },
            {
                "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager.eu-south-1.amazonaws.com"
                    }
                },
                "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Region": "eu-south-1"
                }
            },
            {
                "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
```

```
            "url": "https://license-manager.eu-west-1.amazonaws.com"
          }
        },
        "params":
  {
            "UseFIPS": false,
            "UseDualStack": false,
            "Region": "eu-west-1"
          }
      },
      {
        "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.eu-west-2.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": false,
          "UseDualStack": false,
          "Region": "eu-west-2"
        }
      },
      {
        "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.eu-west-3.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": false,
          "UseDualStack": false,
          "Region": "eu-west-3"
        }

      },
      {
        "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.me-south-1.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": false,
          "UseDualStack": false,
          "Region": "me-south-1"
```

```
        }
      },
      {
        "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager.sa-east-1.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": false,
          "UseDualStack": false,
          "Region": "sa-east-1"
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {

          "endpoint": {
              "url": "https://license-manager.us-east-1.amazonaws.com"
            }
        },
        "params": {
          "UseFIPS": false,
          "UseDualStack": false,
          "Region": "us-east-1"
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
              "url": "https://license-manager-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "UseFIPS": true,
          "UseDualStack": false,
          "Region": "us-east-1"
        }
      },
      {
        "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
              "url": "https://license-manager.us-east-2.amazonaws.com"
          }
```

```
        },

    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "us-east-2"
    }
},
{
    "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-east-2.amazonaws.com"
        }
    },
    "params": {
        "UseFIPS": true,
        "UseDualStack": false,
        "Region": "us-east-2"
    }
},
{
    "documentation": "For region us-west-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-west-1.amazonaws.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "us-west-1"

}
},
{
    "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-west-1.amazonaws.com"
        }
    },
    "params": {
        "UseFIPS": true,
        "UseDualStack": false,
        "Region": "us-west-1"
    }
},
```

```
{
    "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-west-2.amazonaws.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "us-west-2"
    }
},
{
    "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",
    "expect":

{
        "endpoint": {
            "url": "https://license-manager-fips.us-west-2.amazonaws.com"
        }
    },
    "params": {
        "UseFIPS": true,
        "UseDualStack": false,
        "Region": "us-west-2"
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.us-east-1.api.aws"
        }
    },
    "params": {
        "UseFIPS": true,
        "UseDualStack": true,
        "Region": "us-east-1"
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.us-east-1.api.aws"
        }
    },
```

```
    "params": {
        "UseFIPS": false,
        "UseDualStack": true,
        "Region": "us-east-1"
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "cn-north-1"
    }
},
{
    "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "cn-northwest-1"
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "UseFIPS": true,
        "UseDualStack": true,
        "Region": "cn-north-1"
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
```

```
        "endpoint": {
          "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "UseFIPS": true,
        "UseDualStack": false,
        "Region": "cn-north-1"
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled
and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": true,
        "Region": "cn-north-1"
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "us-gov-east-1"
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": true,
        "UseDualStack": false,
        "Region": "us-gov-east-1"
```

```
      }
    },
    {
      "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-gov-west-1.amazonaws.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "us-gov-west-1"
      }
    },
    {
      "documentation": "For region us-gov-west-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
        }
      },
      "params": {

        "UseFIPS": true,
        "UseDualStack": false,
        "Region": "us-gov-west-1"
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "UseFIPS": true,
        "UseDualStack": true,
        "Region": "us-gov-east-1"
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-gov-east-1.api.aws"
        }
```

```
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": true,
        "Region": "us-gov-east-1"
      }


  },
  {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "us-iso-east-1"
      }
  },
  {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
      },
      "params": {
        "UseFIPS": true,
        "UseDualStack": true,
        "Region": "us-iso-east-1"
      }
  },
  {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

    "endpoint": {
          "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "UseFIPS": true,
        "UseDualStack": false,
        "Region": "us-iso-east-1"
      }
  },
  {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": true,
        "Region": "us-iso-east-1"
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params":

{
        "UseFIPS": false,
        "UseDualStack": false,
        "Region": "us-isob-east-1"
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
      },
      "params": {
        "UseFIPS": true,
        "UseDualStack": true,
        "Region": "us-isob-east-1"
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "UseFIPS": true,
        "UseDualStack": false,
        "Region": "us-isob-east-1"
      }
    },
    {
```

```
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
              "error": "DualStack is enabled but this partition does not support DualStack"
          },
          "params": {
              "UseFIPS": false,
              "UseDualStack": true,
              "Region": "us-isob-east-1"
          }
      },
      {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
              "endpoint": {
                  "url": "https://example.com"
              }
          },
          "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Region": "us-east-1",
              "Endpoint": "https://example.com"
          }
      },
      {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
          "expect": {
              "endpoint": {
                  "url": "https://example.com"
              }
          },
          "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
          }
      },
      {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
              "UseFIPS": true,
              "UseDualStack": false,
              "Region": "us-east-1",
              "Endpoint": "https://example.com"
          }
```

```
      },
      {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
          "UseFIPS": false,
          "UseDualStack": true,
          "Region": "us-east-1",
          "Endpoint": "https://example.com"
        }
      },
      {
        "documentation": "Missing region",
        "expect": {
          "error": "Invalid Configuration: Missing Region"
        }
      }
    ],
    "version": "1.0"
}
{
  "testCases": [
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseDualStack": true,
        "UseFIPS": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseDualStack": false,
        "UseFIPS": true
```

```
        }
      },
      {
        "documentation": "For region us-gov-east-1
with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseDualStack": true,
          "UseFIPS": false
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseDualStack": false,
          "UseFIPS": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint":
{
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseDualStack": true,
          "UseFIPS": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
```

```
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseDualStack": false,
        "UseFIPS": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseDualStack": true,
        "UseFIPS": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseDualStack": false,
        "UseFIPS": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
      },
      "params": {
        "Region":
"us-iso-east-1",
        "UseDualStack": true,
        "UseFIPS": true
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseDualStack": false,
        "UseFIPS": true
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseDualStack": true,
        "UseFIPS": false
      }
    },
    {

"documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseDualStack": false,
        "UseFIPS": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseDualStack": true,
        "UseFIPS": true
      }
```

```
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect":
{
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseDualStack": false,
            "UseFIPS": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseDualStack": true,
            "UseFIPS": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseDualStack": false,
            "UseFIPS": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
        },
        "params": {
            "Region": "us-isob-east-1",
```

```
        "UseDualStack": true,
        "UseFIPS": true
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseDualStack": false,
        "UseFIPS": true
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "error": "DualStack is enabled but this partition does not support DualStack"
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseDualStack": true,
        "UseFIPS": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseDualStack": false,
        "UseFIPS": false
      }
    },
    {

      "documentation": "For custom endpoint with fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
```

```
            },
            "params": {
                "Region": "us-east-1",
                "UseDualStack": false,
                "UseFIPS": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
                "Region": "us-east-1",
                "UseDualStack": false,
                "UseFIPS": true,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
                "Region": "us-east-1",
                "UseDualStack": true,
                "UseFIPS": false,
                "Endpoint": "https://example.com"
            }
        }
    ],
    "version": "1.0"
}
{
 "version": "1.0",
 "examples": {
 }
}
{
 "pagination": {
  "ListIdentityProviders": {
   "input_token": "NextToken",
   "output_token": "NextToken",
   "limit_key": "MaxResults",
   "result_key": "IdentityProviderSummaries"
  },
```

```
  "ListInstances": {
   "input_token": "NextToken",
   "output_token": "NextToken",
   "limit_key": "MaxResults",
   "result_key": "InstanceSummaries"
  },
  "ListProductSubscriptions": {
   "input_token": "NextToken",
   "output_token": "NextToken",
   "limit_key": "MaxResults",
   "result_key": "ProductUserSummaries"
  },
  "ListUserAssociations": {
   "input_token": "NextToken",
   "output_token": "NextToken",
   "limit_key": "MaxResults",
   "result_key": "InstanceUserSummaries"
  }
 }
}
{
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": true,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "String"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
      "type": "Boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
```

```
    "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "String"
      }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ],
      "type": "tree",
      "rules": [
        {
          "conditions": [
            {
              "fn": "isSet",
              "argv": [
                {
                  "ref": "Endpoint"
                }
              ]
            }
          ],
          "type": "tree",

    "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
```

```
                    "type": "error"
                },
                {
                    "conditions": [],
                    "type": "tree",
                    "rules": [
                        {
                            "conditions": [
                                {

            "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
                                        },
                                        true
                                    ]
                                }
                            ],
                            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                            "type": "error"
                        },
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": {
                                    "ref": "Endpoint"
                                },
                                "properties":
{},

                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ]
                }
            ]
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
```

```
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        }
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]
                        }
                      ]
                    }
                  ],
                  "type": "tree",
```

```
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
]
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or both",
    "type": "error"
}
]
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true

]
        }
    ],
    "type": "tree",
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
```

```
                    ]
                  }
                ]
              }
            ],

    "type": "tree",
                "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ]
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled but this partition does not support FIPS",
                  "type": "error"
                }
              ]
            },
            {
              "conditions": [
                {

        "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
```

```
                                  {
                                    "fn": "getAttr",
                                    "argv": [
                                      {
                                        "ref": "PartitionResult"
                                      },

                        "supportsDualStack"
                                    ]
                                  }
                                ]
                              }
                            ],
                            "type": "tree",
                            "rules": [
                              {
                                "conditions": [],
                                "endpoint": {
                                  "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
                                "type": "endpoint"
                              }
                            ]
                          },
                          {
                            "conditions": [],
                            "error":
"DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                          }
                        ]
                      },
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-linux-subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ]
                  }
                ]
              }
```

{
 "version":"2.0",
 "metadata":{
  "apiVersion":"2018-05-10",
  "endpointPrefix":"license-manager-user-subscriptions",
  "jsonVersion":"1.1",
  "protocol":"rest-json",
  "serviceFullName":"AWS License Manager User Subscriptions",
  "serviceId":"License Manager User Subscriptions",
  "signatureVersion":"v4",
  "signingName":"license-manager-user-subscriptions",
  "uid":"license-manager-user-subscriptions-2018-05-10"
 },
 "operations":{
  "AssociateUser":{
   "name":"AssociateUser",
   "http":{
    "method":"POST",
    "requestUri":"/user/AssociateUser",
    "responseCode":200

```
      },
     "input":{"shape":"AssociateUserRequest"},
     "output":{"shape":"AssociateUserResponse"},
     "errors":[
       {"shape":"ServiceQuotaExceededException"},
       {"shape":"ConflictException"},
       {"shape":"ValidationException"},
       {"shape":"ThrottlingException"},
       {"shape":"InternalServerException"},
       {"shape":"ResourceNotFoundException"},
        {"shape":"AccessDeniedException"}
     ],
     "idempotent":true
    },
    "DeregisterIdentityProvider":{
     "name":"DeregisterIdentityProvider",
     "http":{
       "method":"POST",
       "requestUri":"/identity-provider/DeregisterIdentityProvider",
       "responseCode":200
      },
     "input":{"shape":"DeregisterIdentityProviderRequest"},
     "output":{"shape":"DeregisterIdentityProviderResponse"},
     "errors":[
       {"shape":"ServiceQuotaExceededException"},
       {"shape":"ConflictException"},
       {"shape":"ValidationException"},
       {"shape":"ThrottlingException"},
       {"shape":"InternalServerException"},
       {"shape":"ResourceNotFoundException"},
       {"shape":"AccessDeniedException"}
     ],
     "idempotent":true
    },
    "DisassociateUser":{
     "name":"DisassociateUser",
     "http":{
       "method":"POST",
       "requestUri":"/user/DisassociateUser",
       "responseCode":200
      },
     "input":{"shape":"DisassociateUserRequest"},
      "output":{"shape":"DisassociateUserResponse"},
     "errors":[
       {"shape":"ServiceQuotaExceededException"},
       {"shape":"ConflictException"},
       {"shape":"ValidationException"},
       {"shape":"ThrottlingException"},
```

```
   {"shape":"InternalServerException"},
   {"shape":"ResourceNotFoundException"},
   {"shape":"AccessDeniedException"}
  ],
  "idempotent":true
},
"ListIdentityProviders":{
  "name":"ListIdentityProviders",
  "http":{
    "method":"POST",
    "requestUri":"/identity-provider/ListIdentityProviders",
    "responseCode":200
  },
  "input":{"shape":"ListIdentityProvidersRequest"},
  "output":{"shape":"ListIdentityProvidersResponse"},
  "errors":[
    {"shape":"ServiceQuotaExceededException"},
    {"shape":"ConflictException"},
    {"shape":"ValidationException"},
    {"shape":"ThrottlingException"},
    {"shape":"InternalServerException"},
    {"shape":"ResourceNotFoundException"},
    {"shape":"AccessDeniedException"}
  ]
},
"ListInstances":{
  "name":"ListInstances",
  "http":{
    "method":"POST",
    "requestUri":"/instance/ListInstances",
    "responseCode":200
  },
  "input":{"shape":"ListInstancesRequest"},
  "output":{"shape":"ListInstancesResponse"},
  "errors":[
    {"shape":"ServiceQuotaExceededException"},
    {"shape":"ConflictException"},
    {"shape":"ValidationException"},
    {"shape":"ThrottlingException"},
    {"shape":"InternalServerException"},
    {"shape":"ResourceNotFoundException"},
    {"shape":"AccessDeniedException"}
  ]
},
"ListProductSubscriptions":{
  "name":"ListProductSubscriptions",
  "http":{
    "method":"POST",
```

   "requestUri":"/user/ListProductSubscriptions",
   "responseCode":200
  },
  "input":{"shape":"ListProductSubscriptionsRequest"},
  "output":{"shape":"ListProductSubscriptionsResponse"},
  "errors":[
   {"shape":"ServiceQuotaExceededException"},
   {"shape":"ConflictException"},
   {"shape":"ValidationException"},
   {"shape":"ThrottlingException"},
   {"shape":"InternalServerException"},
   {"shape":"ResourceNotFoundException"},
   {"shape":"AccessDeniedException"}
  ]
 },
 "ListUserAssociations":{
  "name":"ListUserAssociations",
  "http":{
   "method":"POST",
   "requestUri":"/user/ListUserAssociations",
   "responseCode":200
  },
  "input":{"shape":"ListUserAssociationsRequest"},
  "output":{"shape":"ListUserAssociationsResponse"},
  "errors":[
   {"shape":"ServiceQuotaExceededException"},
   {"shape":"ConflictException"},
   {"shape":"ValidationException"},
   {"shape":"ThrottlingException"},
   {"shape":"InternalServerException"},
   {"shape":"ResourceNotFoundException"},
   {"shape":"AccessDeniedException"}
  ]
 },
 "RegisterIdentityProvider":{
  "name":"RegisterIdentityProvider",
  "http":{
   "method":"POST",
   "requestUri":"/identity-provider/RegisterIdentityProvider",
   "responseCode":200
  },
  "input":{"shape":"RegisterIdentityProviderRequest"},
  "output":{"shape":"RegisterIdentityProviderResponse"},
  "errors":[
   {"shape":"ServiceQuotaExceededException"},
   {"shape":"ConflictException"},
   {"shape":"ValidationException"},
   {"shape":"ThrottlingException"},

```
      {"shape":"InternalServerException"},
      {"shape":"ResourceNotFoundException"},
      {"shape":"AccessDeniedException"}
     ],
     "idempotent":true
    },
    "StartProductSubscription":{
     "name":"StartProductSubscription",
     "http":{
      "method":"POST",
      "requestUri":"/user/StartProductSubscription",
      "responseCode":200
     },
     "input":{"shape":"StartProductSubscriptionRequest"},

     "output":{"shape":"StartProductSubscriptionResponse"},
     "errors":[
      {"shape":"ServiceQuotaExceededException"},
      {"shape":"ConflictException"},
      {"shape":"ValidationException"},
      {"shape":"ThrottlingException"},
      {"shape":"InternalServerException"},
      {"shape":"ResourceNotFoundException"},
      {"shape":"AccessDeniedException"}
     ]
    },
    "StopProductSubscription":{
     "name":"StopProductSubscription",
     "http":{
      "method":"POST",
      "requestUri":"/user/StopProductSubscription",
      "responseCode":200
     },
     "input":{"shape":"StopProductSubscriptionRequest"},
     "output":{"shape":"StopProductSubscriptionResponse"},
     "errors":[
      {"shape":"ServiceQuotaExceededException"},
      {"shape":"ConflictException"},
      {"shape":"ValidationException"},
      {"shape":"ThrottlingException"},
      {"shape":"InternalServerException"},
      {"shape":"ResourceNotFoundException"},

      {"shape":"AccessDeniedException"}
     ]
    },
    "UpdateIdentityProviderSettings":{
     "name":"UpdateIdentityProviderSettings",
```

```
      "http":{
       "method":"POST",
       "requestUri":"/identity-provider/UpdateIdentityProviderSettings",
       "responseCode":200
      },
      "input":{"shape":"UpdateIdentityProviderSettingsRequest"},
      "output":{"shape":"UpdateIdentityProviderSettingsResponse"},
      "errors":[
       {"shape":"ValidationException"},
       {"shape":"ThrottlingException"},
       {"shape":"InternalServerException"},
       {"shape":"AccessDeniedException"}
      ],
      "idempotent":true
     }
    },
    "shapes":{
     "AccessDeniedException":{
      "type":"structure",
      "members":{
       "message":{"shape":"String"}
      },
      "exception":true
     },
     "ActiveDirectoryIdentityProvider":{
      "type":"structure",
      "members":{
       "DirectoryId":{"shape":"String"}
      }
     },
     "AssociateUserRequest":{
      "type":"structure",
      "required":[
       "IdentityProvider",
       "InstanceId",
       "Username"
      ],
      "members":{
       "Domain":{"shape":"String"},
       "IdentityProvider":{"shape":"IdentityProvider"},
       "InstanceId":{"shape":"String"},
       "Username":{"shape":"String"}
      }
     },
     "AssociateUserResponse":{
      "type":"structure",
      "required":["InstanceUserSummary"],
      "members":{
```

```
    "InstanceUserSummary":{"shape":"InstanceUserSummary"}
   }
  },
  "BoxInteger":{
   "type":"integer",
   "box":true
  },
  "ConflictException":{
   "type":"structure",
   "members":{
    "message":{"shape":"String"}
   },
   "exception":true,
   "fault":true
  },
  "DeregisterIdentityProviderRequest":{
   "type":"structure",
   "required":[
    "IdentityProvider",
    "Product"
   ],
   "members":{
    "IdentityProvider":{"shape":"IdentityProvider"},
    "Product":{"shape":"String"}
   }
  },
  "DeregisterIdentityProviderResponse":{
   "type":"structure",
   "required":["IdentityProviderSummary"],
   "members":{
    "IdentityProviderSummary":{"shape":"IdentityProviderSummary"}
   }
  },
  "DisassociateUserRequest":{
   "type":"structure",
   "required":[
    "IdentityProvider",
    "InstanceId",
    "Username"
   ],
   "members":{
    "Domain":{"shape":"String"},
    "IdentityProvider":{"shape":"IdentityProvider"},
    "InstanceId":{"shape":"String"},
    "Username":{"shape":"String"}
   }
  },
  "DisassociateUserResponse":{
```

```
    "type":"structure",
    "required":["InstanceUserSummary"],
    "members":{
      "InstanceUserSummary":{"shape":"InstanceUserSummary"}
    }
  },
  "Filter":{
    "type":"structure",
    "members":{
      "Attribute":{"shape":"String"},
      "Operation":{"shape":"String"},
      "Value":{"shape":"String"}
    }
  },
  "FilterList":{
    "type":"list",
    "member":{"shape":"Filter"}
  },
  "IdentityProvider":{
    "type":"structure",
    "members":{
      "ActiveDirectoryIdentityProvider":{"shape":"ActiveDirectoryIdentityProvider"}
    },
    "union":true
  },
  "IdentityProviderSummary":{
    "type":"structure",
    "required":[
      "IdentityProvider",
      "Product",
      "Settings",
      "Status"
    ],
    "members":{
      "FailureMessage":{"shape":"String"},
      "IdentityProvider":{"shape":"IdentityProvider"},
      "Product":{"shape":"String"},
      "Settings":{"shape":"Settings"},
      "Status":{"shape":"String"}
    }
  },
  "IdentityProviderSummaryList":{
    "type":"list",
    "member":{"shape":"IdentityProviderSummary"}
  },
  "InstanceSummary":{
    "type":"structure",
    "required":[
```

```
      "InstanceId",
      "Products",
      "Status"
     ],
     "members":{
      "InstanceId":{"shape":"String"},
      "LastStatusCheckDate":{"shape":"String"},
       "Products":{"shape":"StringList"},
      "Status":{"shape":"String"},
      "StatusMessage":{"shape":"String"}
     }
    },
    "InstanceSummaryList":{
     "type":"list",
     "member":{"shape":"InstanceSummary"}
    },
    "InstanceUserSummary":{
     "type":"structure",
     "required":[
      "IdentityProvider",
      "InstanceId",
      "Status",
      "Username"
     ],
     "members":{
      "AssociationDate":{"shape":"String"},
      "DisassociationDate":{"shape":"String"},
      "Domain":{"shape":"String"},
      "IdentityProvider":{"shape":"IdentityProvider"},
      "InstanceId":{"shape":"String"},
      "Status":{"shape":"String"},
      "StatusMessage":{"shape":"String"},
      "Username":{"shape":"String"}
     }
    },
    "InstanceUserSummaryList":{
     "type":"list",
     "member":{"shape":"InstanceUserSummary"}
    },
    "InternalServerException":{
     "type":"structure",
     "members":{

     "message":{"shape":"String"}
     },
     "exception":true,
     "fault":true
    },
```

```
"ListIdentityProvidersRequest":{
 "type":"structure",
 "members":{
  "MaxResults":{"shape":"BoxInteger"},
  "NextToken":{"shape":"String"}
 }
},
"ListIdentityProvidersResponse":{
 "type":"structure",
 "required":["IdentityProviderSummaries"],
 "members":{
  "IdentityProviderSummaries":{"shape":"IdentityProviderSummaryList"},
  "NextToken":{"shape":"String"}
 }
},
"ListInstancesRequest":{
 "type":"structure",
 "members":{
  "Filters":{"shape":"FilterList"},
  "MaxResults":{"shape":"BoxInteger"},
  "NextToken":{"shape":"String"}
 }
},
"ListInstancesResponse":{
 "type":"structure",
 "members":{
  "InstanceSummaries":{"shape":"InstanceSummaryList"},
  "NextToken":{"shape":"String"}
 }
},
"ListProductSubscriptionsRequest":{
 "type":"structure",
 "required":[
  "IdentityProvider",
  "Product"
 ],
 "members":{
  "Filters":{"shape":"FilterList"},
  "IdentityProvider":{"shape":"IdentityProvider"},
  "MaxResults":{"shape":"BoxInteger"},
  "NextToken":{"shape":"String"},
  "Product":{"shape":"String"}
 }
},
"ListProductSubscriptionsResponse":{
 "type":"structure",
 "members":{
  "NextToken":{"shape":"String"},
```

```
    "ProductUserSummaries":{"shape":"ProductUserSummaryList"}
   }
  },
  "ListUserAssociationsRequest":{
   "type":"structure",
   "required":[
    "IdentityProvider",
    "InstanceId"
   ],
   "members":{
    "Filters":{"shape":"FilterList"},
    "IdentityProvider":{"shape":"IdentityProvider"},
    "InstanceId":{"shape":"String"},
    "MaxResults":{"shape":"BoxInteger"},
    "NextToken":{"shape":"String"}
   }
  },
  "ListUserAssociationsResponse":{
   "type":"structure",

"members":{
    "InstanceUserSummaries":{"shape":"InstanceUserSummaryList"},
    "NextToken":{"shape":"String"}
   }
  },
  "ProductUserSummary":{
   "type":"structure",
   "required":[
    "IdentityProvider",
    "Product",
    "Status",
    "Username"
   ],
   "members":{
    "Domain":{"shape":"String"},
    "IdentityProvider":{"shape":"IdentityProvider"},
    "Product":{"shape":"String"},
    "Status":{"shape":"String"},
    "StatusMessage":{"shape":"String"},
    "SubscriptionEndDate":{"shape":"String"},
    "SubscriptionStartDate":{"shape":"String"},
    "Username":{"shape":"String"}
   }
  },
  "ProductUserSummaryList":{
   "type":"list",
   "member":{"shape":"ProductUserSummary"}
  },
```

```
"RegisterIdentityProviderRequest":{
 "type":"structure",
 "required":[
  "IdentityProvider",
  "Product"
 ],
 "members":{
  "IdentityProvider":{"shape":"IdentityProvider"},
   "Product":{"shape":"String"},
  "Settings":{"shape":"Settings"}
 }
},
"RegisterIdentityProviderResponse":{
 "type":"structure",
 "required":["IdentityProviderSummary"],
 "members":{
  "IdentityProviderSummary":{"shape":"IdentityProviderSummary"}
 }
},
"ResourceNotFoundException":{
 "type":"structure",
 "members":{
  "message":{"shape":"String"}
 },
 "error":{
  "httpStatusCode":404,
  "senderFault":true
 },
 "exception":true
},
"SecurityGroup":{
 "type":"string",
 "max":200,
 "min":5,
 "pattern":"^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
},
"ServiceQuotaExceededException":{
 "type":"structure",
 "members":{
  "message":{"shape":"String"}
 },
 "exception":true
},
"Settings":{
 "type":"structure",
 "required":[
  "SecurityGroupId",
  "Subnets"
```

```
    ],
    "members":{
      "SecurityGroupId":{"shape":"SecurityGroup"},
      "Subnets":{"shape":"SettingsSubnetsList"}
    }
  },
  "SettingsSubnetsList":{
    "type":"list",
    "member":{"shape":"Subnet"},
    "min":1
  },
  "StartProductSubscriptionRequest":{
    "type":"structure",
    "required":[
      "IdentityProvider",
      "Product",
      "Username"
    ],
    "members":{
      "Domain":{"shape":"String"},
      "IdentityProvider":{"shape":"IdentityProvider"},
      "Product":{"shape":"String"},
      "Username":{"shape":"String"}
    }
  },
  "StartProductSubscriptionResponse":{
    "type":"structure",
    "required":["ProductUserSummary"],
    "members":{
      "ProductUserSummary":{"shape":"ProductUserSummary"}
    }
  },
  "StopProductSubscriptionRequest":{
    "type":"structure",
    "required":[
      "IdentityProvider",
      "Product",
      "Username"
    ],
    "members":{
      "Domain":{"shape":"String"},
      "IdentityProvider":{"shape":"IdentityProvider"},
      "Product":{"shape":"String"},
      "Username":{"shape":"String"}
    }
  },
  "StopProductSubscriptionResponse":{
    "type":"structure",
```

    "required":["ProductUserSummary"],
   "members":{
    "ProductUserSummary":{"shape":"ProductUserSummary"}
   }
  },
  "String":{"type":"string"},
  "StringList":{
   "type":"list",
   "member":{"shape":"String"}
  },
  "Subnet":{
   "type":"string",
   "pattern":"subnet-[a-z0-9]{8,17}"
  },
  "Subnets":{
   "type":"list",
   "member":{"shape":"Subnet"}
  },
  "ThrottlingException":{
   "type":"structure",
   "members":{
    "message":{"shape":"String"}
   },
   "exception":true
  },
  "UpdateIdentityProviderSettingsRequest":{
   "type":"structure",
   "required":[
    "IdentityProvider",
    "Product",
    "UpdateSettings"
   ],
   "members":{
    "IdentityProvider":{"shape":"IdentityProvider"},
    "Product":{"shape":"String"},
    "UpdateSettings":{"shape":"UpdateSettings"}
   }
  },
  "UpdateIdentityProviderSettingsResponse":{
   "type":"structure",
   "required":["IdentityProviderSummary"],
   "members":{
    "IdentityProviderSummary":{"shape":"IdentityProviderSummary"}
   }
  },
  "UpdateSettings":{
   "type":"structure",
   "required":[

      "AddSubnets",
      "RemoveSubnets"
    ],
    "members":{
      "AddSubnets":{"shape":"Subnets"},
      "RemoveSubnets":{"shape":"Subnets"},
      "SecurityGroupId":{"shape":"SecurityGroup"}
    }
  },
  "ValidationException":{
    "type":"structure",
    "members":{
      "message":{"shape":"String"}
    },
    "exception":true
  }
 }
}

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

    "License" shall mean the terms and conditions for use, reproduction,
    and distribution as defined by Sections 1 through 9 of this document.

    "Licensor" shall mean the copyright owner or entity authorized by
    the copyright owner that is granting the License.

    "Legal Entity" shall mean the union of the acting entity and all
    other entities that control, are controlled by, or are under common
    control with that entity. For the purposes of this definition,
    "control" means (i) the power, direct or indirect, to cause the
    direction or management of such entity, whether by contract or
    otherwise, or (ii) ownership of fifty percent (50%) or more of the
    outstanding shares, or (iii) beneficial ownership
of such entity.

    "You" (or "Your") shall mean an individual or Legal Entity
    exercising permissions granted by this License.

    "Source" form shall mean the preferred form for making modifications,
    including but not limited to software source code, documentation
    source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
 and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this

License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

 4. Redistribution. You may reproduce and distribute
copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding
those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
    incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
 distributed under the License is distributed on an "AS IS" BASIS,
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 See the License for the specific language governing permissions and
 limitations under the License.

```
{
 "pagination": {
 }
}
{
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
```

```
        "type": "String"
      },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-
stack, dispatching the request MAY return an error.",
        "type": "Boolean"
      },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured
endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "Boolean"
      },
    "Endpoint": {

  "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "String"
      }
    },
  "rules": [
    {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ],
        "type": "tree",
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
```

```
                    true
                  ]
                }
              ],
              "error": "Invalid Configuration: FIPS and custom
endpoint are not supported",
              "type": "error"
            },
            {
              "conditions": [],
              "type": "tree",
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],

    "endpoint": {
                    "url": {
                      "ref": "Endpoint"
                    },
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }
              ]
            }
          ]
        },
        {
          "conditions": [],
          "type": "tree",
          "rules": [
            {
```

```
"conditions": [
    {
        "fn": "isSet",
        "argv": [
            {
                "ref": "Region"
            }
        ]
    }
],
"type": "tree",
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "type": "tree",
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
```

```
                        "type": "tree",
                        "rules": [
                          {
                            "conditions": [
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  true,
                                  {
                                    "fn": "getAttr",
                                    "argv": [
                                      {
                                        "ref": "PartitionResult"
                                      },
                                      "supportsFIPS"
                                    ]
                                  }
                                ]
                              },
                              {
                                "fn": "booleanEquals",
                                "argv": [

                                  true,
                                  {
                                    "fn": "getAttr",
                                    "argv": [
                                      {
                                        "ref": "PartitionResult"
                                      },
                                      "supportsDualStack"
                                    ]
                                  }
                                ]
                              }
                            ],
                            "type": "tree",
                            "rules": [
                              {

                                "conditions": [],
                                "type": "tree",
                                "rules": [
                                  {
                                    "conditions": [],
                                    "endpoint": {
                                      "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
```

```
                                        "properties": {},
                                        "headers": {}
                                      },
                                      "type": "endpoint"
                                    }
                                  ]

                  }
                            ]
                          },
                          {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                            "type": "error"
                          }
                        ]
                      },
                      {
                        "conditions": [
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              {
                                "ref": "UseFIPS"
                              },
                              true

                            ]
                          }
                        ],
                        "type": "tree",
                        "rules": [
                          {
                            "conditions": [
                              {
                                "fn": "booleanEquals",
                                "argv": [
                                  true,
                                  {
                                    "fn": "getAttr",
                                    "argv": [
                                      {
                                        "ref": "PartitionResult"

                  },

                                      "supportsFIPS"
                                    ]
```

```
                                            }
                                        ]
                                    }
                                ],
                                "type": "tree",
                                "rules": [
                                    {
                                        "conditions": [],
                                        "type": "tree",
                                        "rules": [
                                            {
                                                "conditions": [],
                                                "endpoint": {
                                                    "url":
"https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
                                                    "properties": {},
                                                    "headers": {}
                                                },
                                                "type": "endpoint"
                                            }
                                        ]
                                    }
                                ]
                            },
                            {
                                "conditions": [],
                                "error": "FIPS is enabled but this partition does not support FIPS",
                                "type": "error"
                            }
                        ]

                    },
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
                                        },
                                        true
                                    ]
                                }
                            ],
                            "type": "tree",
                            "rules": [
                                {
                                    "conditions": [
```

```
                              {
                                 "fn": "booleanEquals",
                                 "argv":
[
                                      true,
                                    {
                                       "fn": "getAttr",
                                       "argv": [
                                          {
                                             "ref": "PartitionResult"
                                          },
                                          "supportsDualStack"
                                       ]
                                    }
                                 ]
                              }
                           ],
                           "type": "tree",
                           "rules": [

             {
                                 "conditions": [],
                                 "type": "tree",
                                 "rules": [
                                    {
                                       "conditions": [],
                                       "endpoint": {
                                          "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                          "properties": {},
                                          "headers": {}
                                       },
                                       "type": "endpoint"
                                    }
                                 ]

                }
                    ]
                        },
                        {
                           "conditions": [],
                           "error": "DualStack is enabled but this partition does not support DualStack",
                           "type": "error"
                        }
                     ]
                  },
                  {
                     "conditions": [],
```

```
                    "type": "tree",
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ]
                  }
                ]
              }
            ]
          },
          {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
          }
        ]
      }
    ]
  }
```

1.38 lca-aws-services-log-collector 24.2.0

1.38.1 Available under license :

APPDYNAMICS

END USER LICENSE AGREEMENT

By accessing the Software herein, you (and the organization you represent) ("You") acknowledge and agree that the use

of the Software and open source software are governed by (1) the End User License Agreement found at

https://www.cisco.com/c/en/us/about/legal/cloud-and-software/end_user_license_agreement.html and the applicable Product

Specific Terms found at https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html or (2) any

other superseding agreement between AppDynamics, or its parent company Cisco Systems, Inc., as applicable, and You.

References to End User in such agreements shall mean You.

AppDynamics Proprietary and Confidential * Revision 2023.04